The Making of Google Earth VR

Dominik P. Käser
Google, Inc.
dpkay@google.com

Evan Parker
Google, Inc.
eparker@google.com

Adam Glazier
Google, Inc.
aglazier@google.com

Mike Podwal
Google, Inc.
mpod@google.com

Matt Seegmiller
Google, Inc.
mss@google.com

Chun-Po Wang
Google, Inc.
cpwang@google.com

Per Karlsson
Google, Inc.
perk@google.com

Nadav Ashkenazi
Google, Inc.
nadavash@google.com

Joanna Kim
Google, Inc.
joki@google.com

Andre Le
Google, Inc.
andrele@google.com

Matthias Bühlmann
Mabulous GmbH
matthias.buehlmann@mabulous.com

Joshua Moshier
Thornapple Music
jmoshiermusic@gmail.com

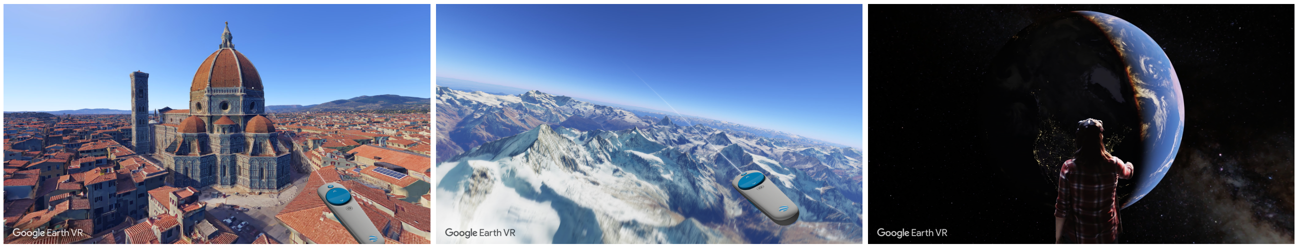

Figure 1: Florence's cathedral of Santa Maria, the Swiss Alps, and the Earth, as featured in Google Earth VR.

ABSTRACT

One of the great promises of virtual reality is that it can allow people to visit places in the world that they might otherwise be unable to. Since the recent renaissance of virtual reality, content creators have exercised various techniques such as 360-degree cameras and photogrammetry to make this promise come true.

At Google, we spent more than 10 years capturing every part of the world as part of the Google Earth project. The result is a rich 3D mesh that contains trillions of triangles [Kontkanen and Parker 2014] and as such is predestined to be a good data source for VR content. In [Kaeser and Buehlmann 2016] we discussed some of our early experiments with bringing Google Earth to virtual reality, but without a focus on developing a product. Following these experiments, we worked extensively to create a well-rounded product, *Google Earth VR*, which we eventually launched to the world in November 2016. Google Earth VR quickly became one of the most actively used VR applications in the market and has won several awards since.

This talk discusses the journey of the Google Earth VR project from its early prototypes to its final launched stage.

CCS CONCEPTS

- **Computing methodologies → Virtual reality**;

KEYWORDS

geospatial visualization, virtual reality, real-time rendering

ACM Reference format:
Dominik P. Käser, Evan Parker, Adam Glazier, Mike Podwal, Matt Seegmiller, Chun-Po Wang, Per Karlsson, Nadav Ashkenazi, Joanna Kim, Andre Le, Matthias Bühlmann, and Joshua Moshier. 2017. The Making of Google Earth VR. In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
https://doi.org/10.1145/3084363.3085094

1 NAVIGATION

It is very hard to design a good UI system for moving through environments in VR. While there has been lots of experimentation, many navigation UI systems in VR cause motion sickness, are unintuitive, or result in a loss of context for how you moved from point A to B. In most VR systems, users are given input controllers whose signals can be mapped to navigation operations such as teleporting or continuous flying forward and backward. In addition, with the advent of positional tracking systems, users are able to move up to a few meters within a tracked volume. Our planet has a radius

---

SIGGRAPH '17 Talks, July 30 - August 03, 2017, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5008-2/17/07.
https://doi.org/10.1145/3084363.3085094

of thousands of kilometers, though, so we have the challenge of mapping real world to virtual world distances. In addition, we want to preserve context for users across navigation operations while preventing them from becoming nauseous

After iterating extensively on 3D navigation techniques, we settled on two methods that tested well on users:

- **3D Cone Drag** allows users to drag the landscape to a new position while ensuring that their feet are always on the planet’s surface. We’ll discuss our final solution and how we resolved issues that arose from it (e.g. near-parallel intersections and nauseating movements).
- **Scale-and-fly** allows users to fly at adaptive speeds while ensuring that their feet are always on the planet’s surface. We’ll discuss how it complements dragging and how we prevent users from becoming nauseous while moving by temporarily reducing their field of view using a technique we call Tunnel Vision.
- **Rotation** allows people with limited accessibility or limited tracking to rotate the virtual world. We’ll discuss the different options explored, our findings and where we ended up.
- **Teleport** allows users to instantly move to a new location on the planet. We’ll discuss how teleport is used differently across the menu, mini-globe and search.
- **Search** allows users to search for locations spanning large areas like cities and specific points of interest like addresses and landmarks. We’ll discuss various challenges, from inputting the search to ending up at a search result.

To prevent users from becoming nauseous, we temporarily reduce the user’s field of view during movement and display a representation of the real world in the periphery. We call this technique *Tunnel Vision* and conducted extensive user studies on it.

2 SOUND AND HAPTIC DESIGN

Good VR design is not limited to visuals – it is key to build a multisensory experience where visuals, sounds and haptics all tell one cohesive story and augment each other. During the creation of Google Earth VR, we kept asking ourselves how it would feel to hold the entire planet in your hands or what it would sound like do spin it around using one’s own manual force. These conversations informed just about any user-visible operation we implemented: when grabbing the planet using the 3D Cone Drag gesture, for example, the Earth exhibits static friction that manifests itself in haptics and dynamic sounds.

Google Earth VR also features a dynamically changing music track, along with ambisonic environmental sounds: we place birds near the planet surface that are audible when users are at small scales. As users scale themselves up, the birds fade and give way to wind and eventually stratospheric noise. As users fly around, the wind noise changes its timbre and directionality.

3 VISUAL AND PRODUCT DESIGN

When we started prototyping Google Earth VR, we knew that we wanted to build a product that includes all of Google Earth’s imagery with dynamic lighting, features Google’s visual language Material Design and yet feels native to the platform (e.g. Steam) we ship on. Material Design only existed as a 2D specification for Google’s client applications, and we will discuss how we took it to VR as we created our virtual control elements and menus.

When it comes to product design, we wanted to both build an open-world applications that allows users to lose themselves in the experience for an indefinite amount of time, yet we wanted to structure the experience from the first second such that users always had a "next thing" to do. We treated the first 5 minutes in the experience like a nonlinear story: we initially take users on a world tour which culminates in a 8-step tutorial and finally we present a few dozens of our favorite places to users. At any time, users can decide to skip a part or abort the sequence entirely, to which the application (visuals, sound and haptics) has to respond gracefully.

4 PERFORMANCE

Google Earth VR renders a huge data set consisting of trillions of triangles. In [Kontkanen and Parker 2014] we have shown how this data is preprocessed into levels of detail, segmented into cells and stored in the nodes of a data structure similar to an oct-tree. Client applications then fetch adaptive geometric detail from the server as necessary.

While this method enables us to display the data set in real-time, in VR we face higher performance requirements than ever before. In [Kaeser and Buehlmann 2016] we discussed some of our early experiments for changes that were necessary to achieve the consistent 90fps that VR demands. During the performance section of this presentation, we will give an summary of our optimization work that led to a shipping product using OpenGL on Windows. This work includes reordering GPU submission commands, draw call batching, more precise timing and working with driver vendors to diagnose spurious stalls.

In addition to frame rate, we also found that scene resolution time (the time it takes to download and fully display a place on the Earth) is key to a good experience. In traditional 2D applications, the virtual camera usually stops between touch/mouse based camera interactions and the rendering engine has time to download the required data. In VR, the camera is tied to the user’s head which is always moving, so we don’t have this luxury and had to get more creative. We will discuss how we addressed scene resolution time even on slower internet connections.

ACKNOWLEDGMENTS

Google Earth VR stands on the shoulders of giants and is based on years of lessons learned in the Google Earth project. The algorithms are intimately tied to the serving infrastructure built by the backend teams. In short, every triangle the client renders is the result of many teams working on data acquisition, computer vision, system architecture, and pipeline maintenance. We want to particularly thank John Anderson and John Rohlf for their crucial rendering optimization work, and Marranda Pousard for her meticulous quality assurance.

REFERENCES

Dominik Kaeser and Matthias Buehlmann. 2016. Bringing Google Earth to Virtual Reality. In *ACM SIGGRAPH 2016 Talks*. ACM, 44.

Janne Kontkanen and Evan Parker. 2014. Earth in Google Maps: rendering trillions of triangles in JavaScript. In *ACM SIGGRAPH 2014 Talks*. ACM, 44.