Rendering in Virtual Reality

Toomas Tamm
What is virtual reality

The computer-generated simulation of a three-dimensional image or environment that can be interacted with in a seemingly real or physical way by a person using special electronic equipment, such as a helmet with a screen inside or gloves fitted with sensors.
Short History

- 1893: First stereoscopic photos & viewers
- 1929: First flight simulator
- 1930: VR goggles in science fiction
- 1950: Sensorama
- 1987: Virtual reality term
- 1990s: First consumer VR devices
- 2012: Oculus
- 2012+: A lot of VR devices

*timeline not in linear scale*
**Immersion & Presence**

- Immersion is user’s engagement with a VR (virtual reality) system that results with being in a flow state. Immersion to VR systems mainly depends on sensory immersion, which is defined as “the degree which the range of sensory channel is engaged by the virtual simulation” (Kim and Biocca 2018).

- Presence is defined as one’s sense of being in the virtual world. The illusion is perceptual but not cognitive, as the perceptual system identifies the events and objects and the brain-body system automatically reacts to the changes in the environment, while cognitive system slowly responds with a conclusion of what the person experiences is an illusion. (Slater 2018).
How do we achieve them?

- Movement and accurate tracking
  - $< \frac{1}{4}$ degrees of rotation, accuracy $< 1$mm
  - Ideally 6 degrees of freedom

- High framerates and low screen persistence
  - $> 80$fps by application
  - Screen persistence $< 3$ms

- High resolution
  - $> 1 \times 1$k per eye

- Large field of view
  - $> 110$ degrees field of view
Some VR device specs

<table>
<thead>
<tr>
<th>Picture</th>
<th>Varjo VR-1</th>
<th>Oculus Rift CV1</th>
<th>Oculus Quest</th>
<th>HTC Vive Pro</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="VR-1" /></td>
<td><img src="image2.png" alt="Oculus Rift CV1" /></td>
<td><img src="image3.png" alt="Oculus Quest" /></td>
<td><img src="image4.png" alt="HTC Vive Pro" /></td>
<td></td>
</tr>
<tr>
<td>Screen resolution</td>
<td>Two 1920x1080 micro-OLED, Two 1440x1660 AMOLED</td>
<td>Two 1080x1200 OLED</td>
<td>Two 1660x1440 OLED</td>
<td>Two 1440x1660 AMOLED</td>
</tr>
<tr>
<td>Refresh rate</td>
<td>60Hz/90hz</td>
<td>90Hz</td>
<td>72Hz</td>
<td>90Hz</td>
</tr>
</tbody>
</table>

(Paal 2021)
What happens if we don’t?

- Motion sickness
- Discomfort
- Nausea
- Dizziness
- Headaches
- Etc

Bad conditions don’t have to last for long, a few seconds is enough.
Why is it hard to achieve?

● 720p @ 30 Hz: 27 million pixels/sec
● 1080p @ 60 Hz: 124 million pixels/sec
● 30” Monitor 2560x1600 @ 60 Hz: 245 million pixels/sec
● 4k Monitor 4096x2160 @ 30 Hz: 265 million pixels/sec
● VR 1512x1680x2 @ 90 Hz: 457 million pixels/sec

3-4x more to render
So how do we do it?
Stereo Rendering
Lenses

Without Lenses

With Lenses

Small FOV

Large FOV
Lens distortion

Image you see  Lens  Screen Image
Barrel distortion

Image you see  Lens  Screen Image
Warped image
Stencil Mesh

- Lenses are round and displays are rectangular
- Won’t see the full display
- So let’s stencil out what we can’t see
Regular Stereo Rendering
Stencil Mesh
Warped image
Warped with Stencil Mesh
Multi-Resolution Rendering/Fixed Foveated Rendering

- Barrel distortion on the image means some pixels get lost
- So we do not have to render corners & edges at same resolution
Foveated Rendering

- Not possible in every headset
- Because you need to be able to track eye position
- Needs low latency
  - Eye saccade time is 20-40ms
Normal VR rendering
Single-Pass Rendering
What if performance still sucks?
Adaptive quality/Adaptive performance

- Dynamically changing rendering settings to maintain performance
- Can
  - Lower resolution
  - Do less antialiasing
  - Increase (fixed) foveated rendering
  - Scale physics/decals/shadows etc..
- When the framerate is low or device is close to throttling
- Or the opposite when you have resources left over

Just don’t go too far, text etc. may become difficult to read/see.
Timewarp

- Reproject already rendered frame
- Doesn’t actually improve framerate
- Lowers perceived latency
Asynchronous Timewarp/Interleaved Reprojection

- Like timewarp
- But we prepare the reprojected frame ahead of time
- Can be used to compensate for dropped frames
- Only accounts for head rotation
Asynchronous Spacewarp/Motion Smoothing

- Reprojects previous frame
- But also accounts for motion in previous frames
- Needs depth buffer from game engine
- Runs at half the framerate (90 -> 45)

DEMO:

https://drive.google.com/file/d/1ISONJxTz3ThsKSoYwm7IP3sAuYN5rLyg/view
Thanks for listening
Sources

https://cgvr.cs.ut.ee/avutigraafika-terminid/
https://www.vrs.org.uk/virtual-reality/history.html
https://www.reddit.com/r/oculus/comments/66nfap/made_a_pic_that_shows_how_foveated_rendering/
https://uploadvr.com/reprojection-explained/
https://www.youtube.com/watch?v=f8piCZz0p-Y
https://drive.google.com/file/d/1lSONJxTz3ThsKSoYwm7IP3sAuYN5rLyg/view
https://www.oed.com/
https://link.springer.com/chapter/10.1007/978-3-319-91584-5_8
https://link.springer.com/referenceworkentry/10.1007/978-3-319-08234-9_162-1