Diving into VR World with Oculus

Homin Lee
Software Engineer at Oculus

GAME DEVELOPERS CONFERENCE™ CHINA
SHANGHAI INTERNATIONAL CONVENTION CENTER
SHANGHAI, CHINA · OCTOBER 19–21, 2014

Topics

- Who is Oculus
- Oculus Rift DK2
- Positional Tracking
- SDK
- Latency
- Roadmap

1. Who is Oculus

Movie

1. Oculus is…

- Palmer Luckey & John Carmack “homebrew” prototype at E3 2012
- Oculus VR founded mid 2012
- Successful Kickstarter campaign Sept 2012 (2.4 million)
- First 10k dev kits shipped March 2013
- Over 50,000 dev kits shipped to date
- Over 60,000 developers on Oculus dev portal
- 100s of games in development for the Rift
- 50+ top-tier show awards, best of CES 2013, E3 2013, CES 2014
- Acquired by Facebook
- Started DK2 shipping on August 2014
- Announced mobile VR kit, Gear VR
- Announced prototype of next version, Crescent bay at Oculus Connect 2014

What’s the most important things in VR

- Low Persistence
- Latency
- Realistic

If it doesn't satisfy them?? Probably

2. Oculus Rift DK2

- 100 Degree Fov • 1080p OLED Screen, 960x1080 per eye
- 75hz Low Persistence refresh rate
- Full Persistence @ 60, 72 or 75 Hz
- 1000Hz IMU(**inertial measurement unit**) tracks orientation and assists position
- Sensor Fusion SW combines IMU and camera data for 6DOF tracking

Crescent Bay – Next prototype of DK2

- Better Screen Resolution
- 90hz Low Persistence refresh rate
- 360-degree tracking
- Improved weight
- Integrated audio

Gear VR

- 1440p OLED Screen
- 60hz refresh Rate
- Working with Samsung Galaxy Note4 for now
- Wireless

3. Positional Tracking

- External camera looking at the user.
- 72H x 52V Degrees FOV
- 0.5m – 2.5 meters tracking range
- Placed on the desk or looking down at the user
- Includes a tripod screw mount

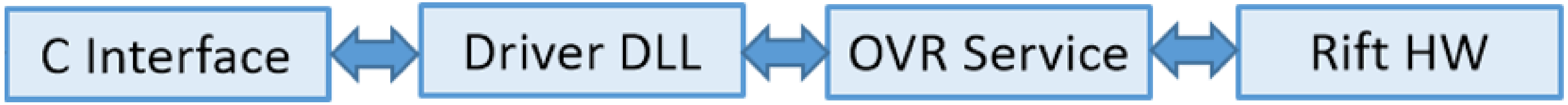
Positional Tracking

- Relies on IR built into the Rift on all sid
- Supports swaying side to side, front and ba
- Tracking works while looking up and down
- Player can turn more than 110 degrees to each side

4. Oculus SDK

- Features of SDK 0.4.x
  - **Support DK1 & DK2**
    - New C Interface; recompile the game with new SDK
  - **Positional tracking**
    - Position origin is currently 1 meter from camera
    - SDK reports SensorState with predicted Pose state
    - Includes orientation, position and derivatives
    - Sets flags when out of tracking range
    - Falls back on head model
  - **Direct-to-Rift and Extended Mode**
  - **OVR Configuration Utility**

Where OVR Software Stack is Heading



- C Interface: Easy to bind from other languages
- Driver DLL: Automatically support changes in HW and functionality
- OVR Service: Rift sharing and VRs transitions across apps

SDK Rendering vs. Game Rendering

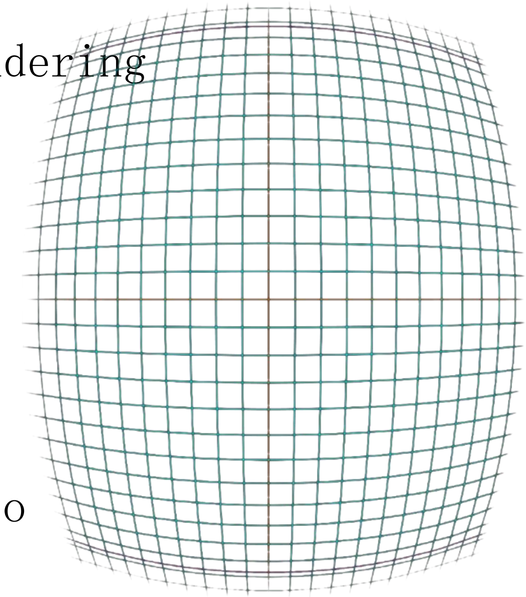
0.2 SDK didn’t do any rendering

Only provided parameters needed for proper rendering

New rendering backend in SDK 0.4

Takes over critical rendering features

Game(App) layer gives SDK L & R eye textures to ovrHmd_EndFrame()

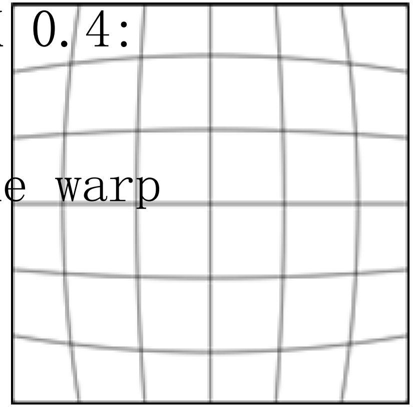


SDK Rendering vs. Game Rendering

- General Work Flow
  - ovrHmd_CreateDistortionMesh
    Transforms the image through UVs
    More efficient to render then pixel shader
    Gives Oculus flexibility to change distortion
  - ovrHmd_BeginFrame
  - ovrHmd_GetEyePoses
  - Stereo rendering based on EyeRenderPose (Game Scene Render)
  - ovrHmd_EndFrame

SDK Rendering vs. Game Rendering

Finishes rendering with the following features on SDK 0.4:



- Barrel distortion with chromatic aberration & time warp
- Internal latency testing & dynamic prediction
- Low-latency v-sync and flip (even better with direct-to-rift)
- Health & Safety Warning

SDK Integration

- **Easy to integrate:**
  - No shaders, meshes to create
  - Pass device/system pointers and eye texture(s)
  - Support OpenGL and D3D9/10/11
  - Must re-apply rendering state for next frame

- **Benefits:**
  - Better compatibility with future Oculus HW and Features
  - Reduces graphics setup bugs
  - Support low-latency driver screen access such as front buffer rendering, etc.
  - Automatically support overlays: latency testing, camera guides, debug data,

Supporting Game Engine

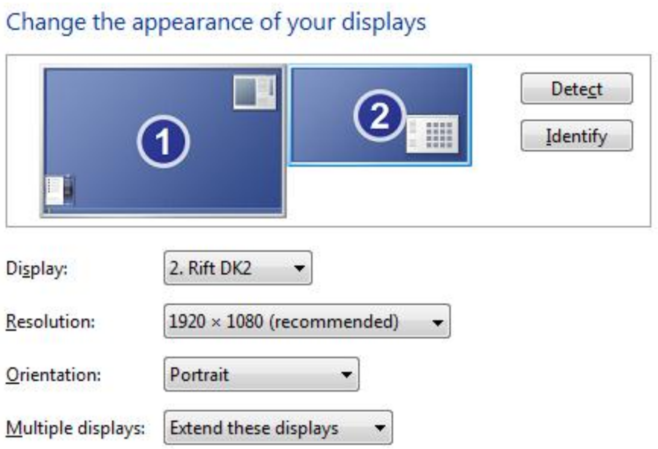
Unity 3D, Unreal Engine 3 & 4 use SDK rendering

In some cases, what you think is unsupported by our SDK, might actually be supported, but we still want to know!

Ping us. We will listen.

Extended Mode

- Headset shows up as an OS Display
- Application must place a window on Rift monit
- Icons and Windows in a wrong place
- Windows compositor handles Present
- There is usually at least one extra frame of latency
- More latency if no CPU & GPU sync is done

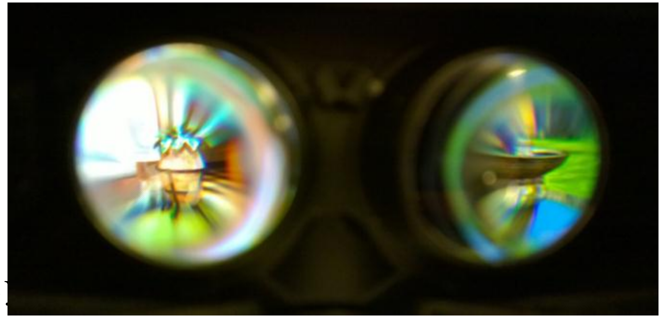

Change the appearance of your displays

Display: 2. Rift DK2
Resolution: 1920 × 1080 (recommended)
Orientation: Portrait
Multiple displays: Extend these displays

Direct To Rift

Outputs to Rift without Display being a part of the desktop

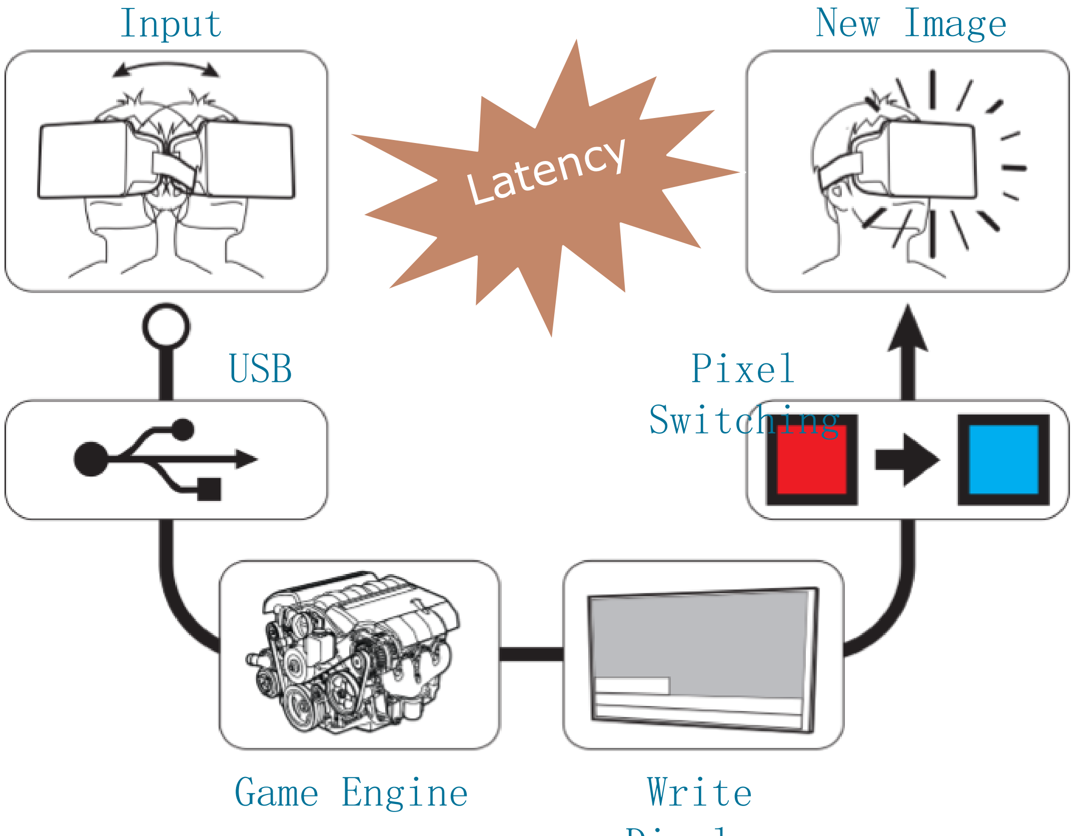
- Headset not seen by the OS
- Avoid jumping windows and icons
- Decouples Rift v-sync from OS compositor
- Avoids extra GPU buffering for minimum latenc
- Use ovrHmd_AttachToWindow
- Window swap chain output is directed to the Rift



Hope to see Direct Mode become the longer term solution

5. Latency

- Motion-to-photon delay
- Multiple stages
  - Motion
  - Sensor delay
  - Processing & fusion
  - Rendering
  - Scanout
  - Transmission
  - Pixel change time
  - Pixel persistence

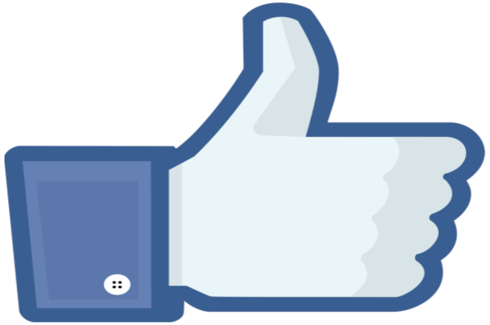


5. Latency

Keeping latency low is crucial for a solid VR experience

Goal is < 20 ms, and hopefully close to 5 ms

We’ re almost there! (stay tuned for some numbers)



Rendering Latency - TimeWarp

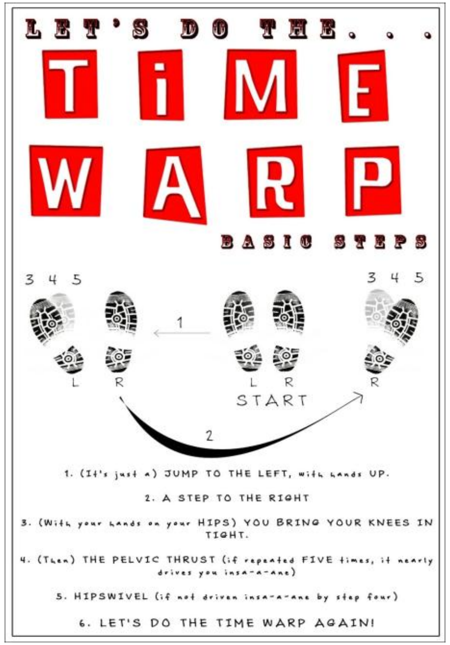
Re-projects rendering for a later point in time

Done simultaneously with distortion

educes perceived latency

Accounts for DK2 rolling shutter

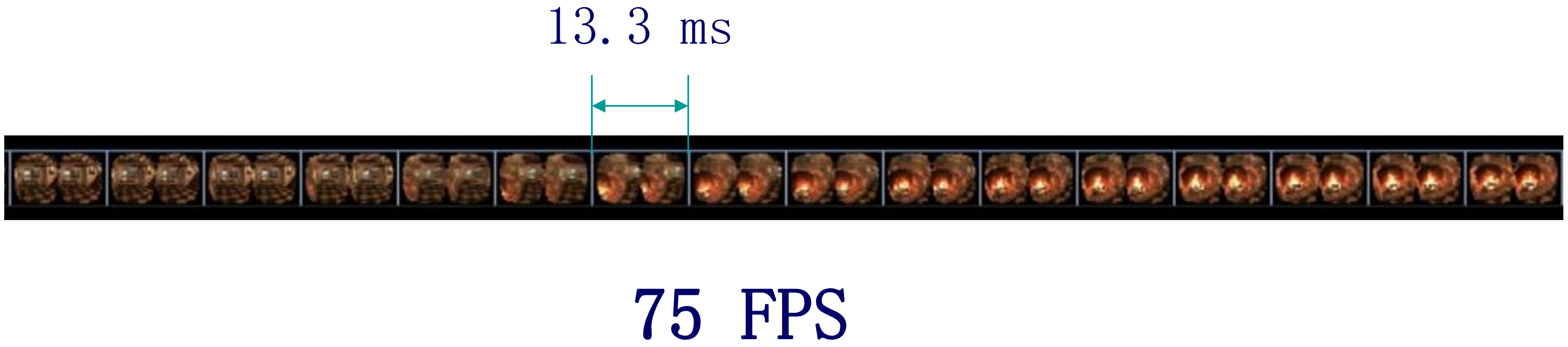
0.4 SDK handles orientation, positional possible


LET'S DO THE. . .
TIME
WARP
BASIC STEPS
3 4 5
3 4 5
1
L R
L R
R
START
2
1. (It's just a) JUMP TO THE LEFT, with hands UP.
2. A STEP TO THE RIGHT
3. (With your hands on your HIPS) YOU BRING YOUR KNEES IN TIGHT.
4. (Then) THE PELVIC THRUST (if repeated FIVE times, it nearly drives you insa-a-ane)
5. HIPSWIVEL (if not driven insa-a-ane by step four)
6. LET'S DO THE TIME WARP AGAIN!

Reducing Latency - TimeWarp

Any other way to apply sensor before the end of the frame?

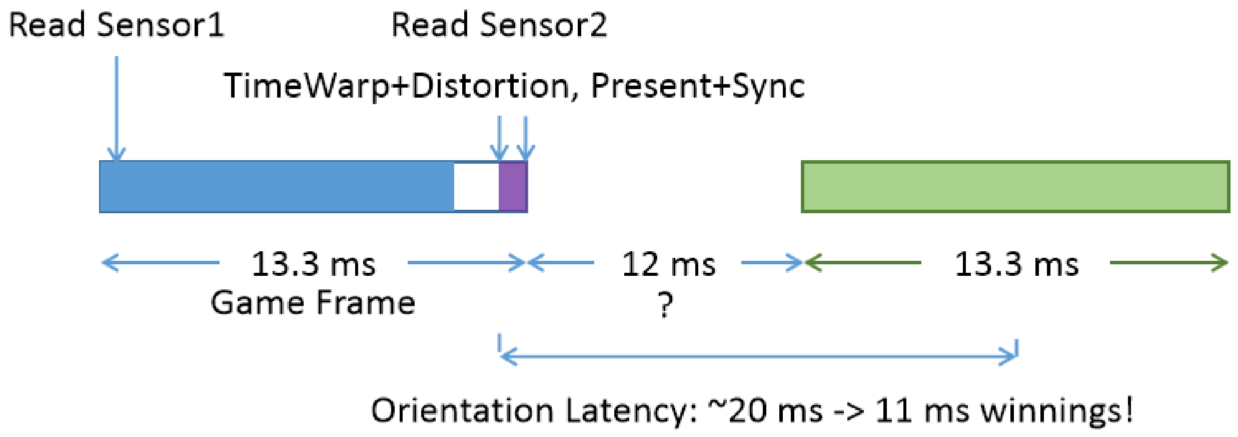
TimeWarp - Projected rendering – Pioneered by John



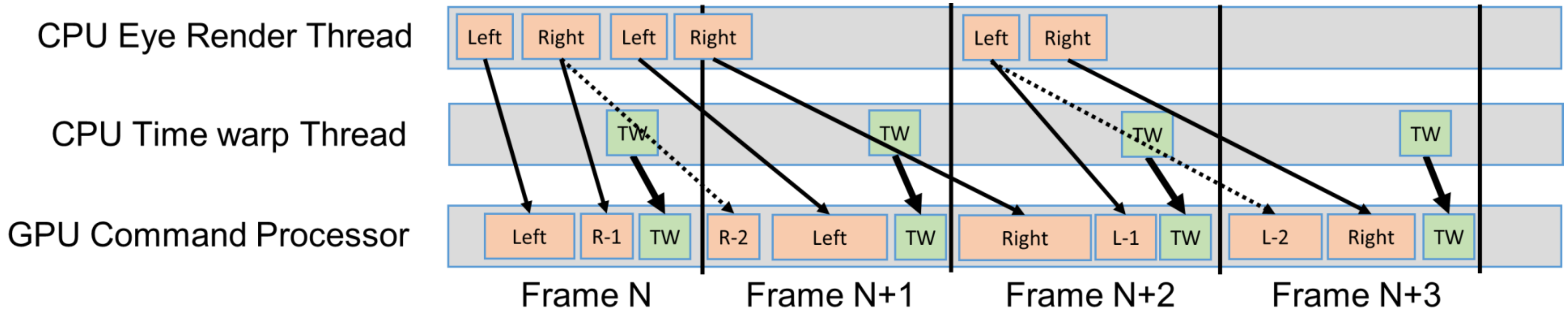
Reducing Latency - TimeWarp

Any other way to apply sensor before the end of the frame?

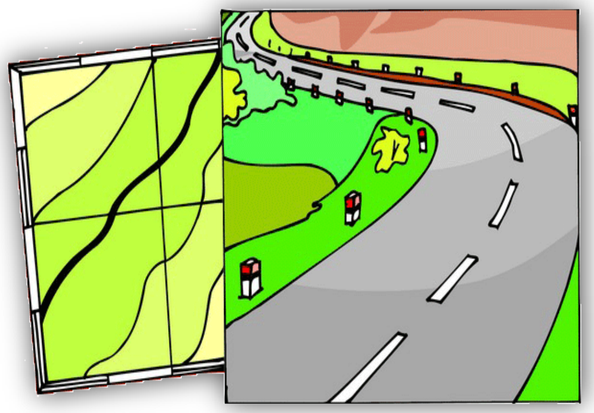
TimeWarp - Projected rendering - Pioneered by John


Read Sensor1
Read Sensor2
TimeWarp+Distortion, Present+Sync
13.3 ms
Game Frame
12 ms
?
13.3 ms
Orientation Latency: ~20 ms -> 11 ms winnings!

Reducing Latency - TimeWarp



6. SDK Roadmap



Driver Improvements & Fixes

- Multi-Monitor fixes to v-sync related latency & judder
- Reduce post-present latency
- Handle Optimus and multi-GPU scenarios

Linux Support

Asynchronous/Adaptive Time warp

- Reduce Judder by generating extra frames when needed

Time warp Layers

- Handle Cockpit and Overlays separately from the scene