

Oculus带你进入 虚拟现实世界

Homin Lee

软件工程师 | OculusVR



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

话题:

- Oculus是什么
- Oculus Rift DK2
- 位置跟踪
- SDK
- 延迟
- 路线图

1. Oculus是什么



Movie

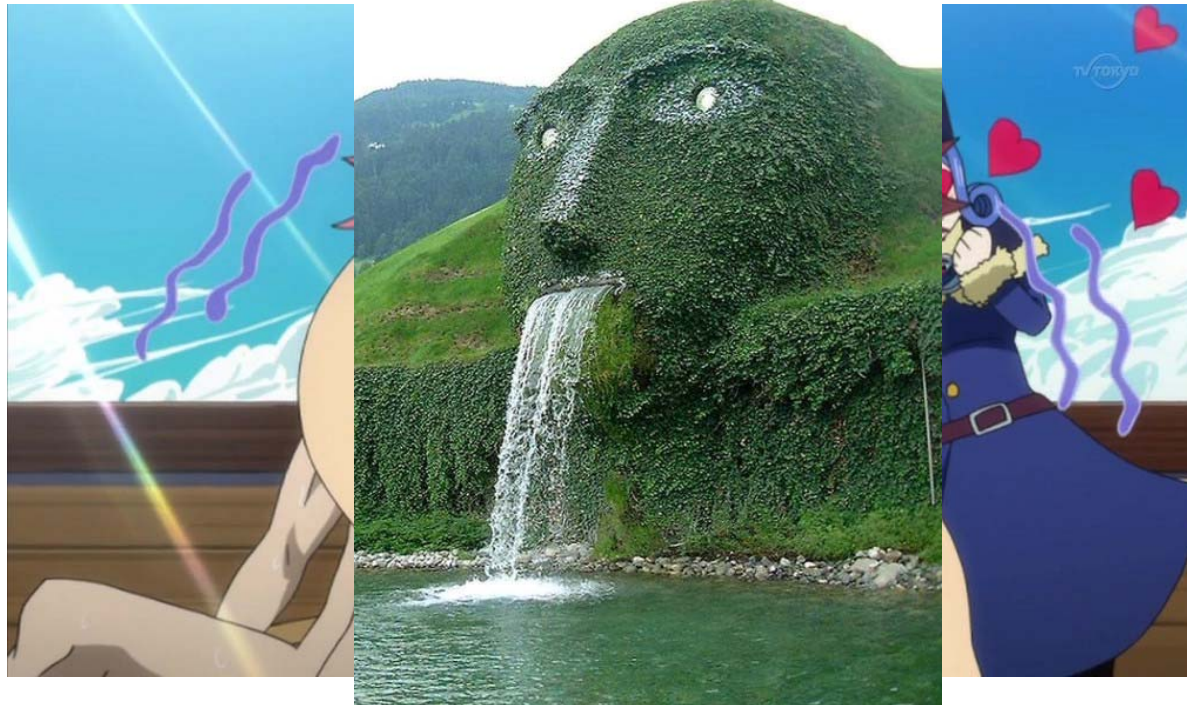
1. Oculus...

- Palmer Luckey和John Carmack在2012E3游戏展上的推出的原型
- Oculus VR成立于2012年年中
- 2012年9月成功地在Kickstarter网站平台上筹资 (240 million)
- 2013年3月, 发售首批1万套dev kit穿戴设备
- 迄今dev kit穿戴设备发售量超过5万套
- 超过60,000名开发者使用 Oculus开发门户
- 正在为Rift开发数百款游戏
- 50多个顶级展会奖项, 获得CES 2013、E3 2013、CES 2014最佳
- 被Facebook收购
- 2014年8月开始发售DK2
- 移动虚拟现实装备Gear VR
- 在Oculus Connect 2014上推出新一代原型Crescent Bay

虚拟现实中最重要哪些方面

- 短余辉
- 延迟
- 现实

如何这未能满足他们?? 也许



2. Oculus Rift DK2

- 100度Fov • 1080p OLED显示屏, 每只眼分辨率 960x1080
- 75hz 短余辉刷新率
- 在60、72或75 Hz时, 全余辉
- 1000Hz惯性测量单元 (IMU) 跟踪方向并协助定位
- Fusion SW传感器将IMU摄像头数据, 以便进行6DOF跟踪



Crescent Bay – DK2的下一个原型

- 显示屏分辨率提高
- 90hz 短余辉刷新率
- 360度跟踪
- 更轻
- 集成音频



Gear VR

- 1440p OLED显示屏
- 60hz刷新率
- 与三星合作
目前采用Galaxy Note4显示屏
- 无线



3. 位置跟踪

- 外部摄像头监视使用者
- 72H x 52V 度 FOV
- 0.5 - 2.5米跟踪范围
- 置于桌上, 或从上方监视使用者
- 包括一个三脚架螺旋底座



位置跟踪

- 依赖于在各面植入Rift的IR
- 支持左右、前后摇晃
- 当向上、向下看时, 跟踪可以进行
- 玩家可向每侧转110度以上



4. Oculus SDK

- SDK 0.4.x的特色

- 支持DK1和DK2

- 新款C接口；用新款SDK来对游戏进行重新编译

- 位置跟踪

- 位置原点目前距离摄像头1米

- SDK用预测的Pose状态来报告传感器状态

- 包括方向、位置和导数

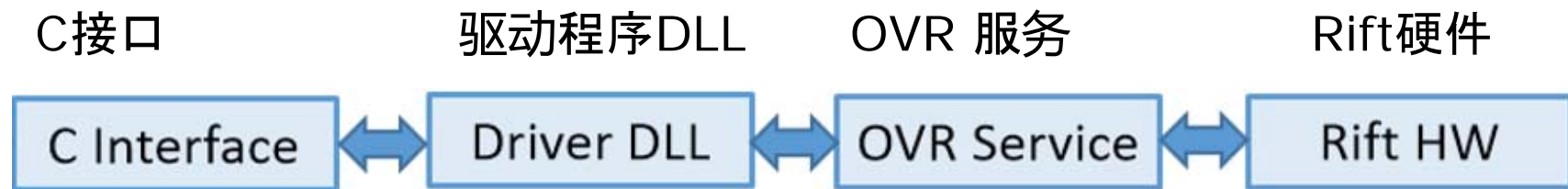
- 超出跟踪范围时, 给出旗标提示

- 依靠头部模型

- Direct-to-Rift和扩展模式

- OVR配置工具

OVR软件堆栈走向何处



- C接口：容易与其他语言连接
- 驱动程序DLL：自动支持硬件和功能的变更
- OVR服务：在各应用之间的Rift分享和虚拟现实转换

SDK渲染与游戏渲染的比较

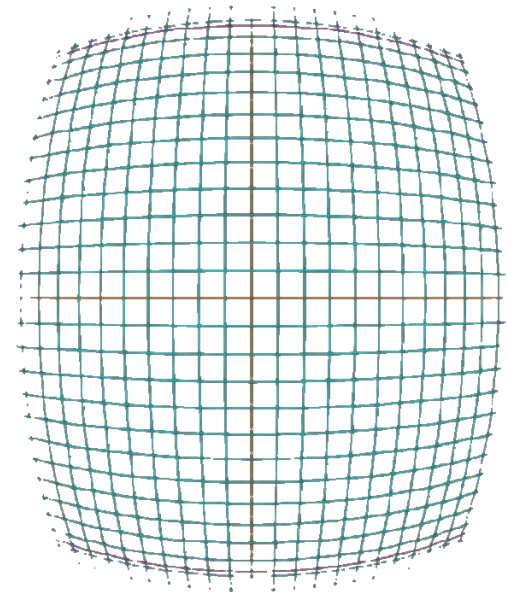
0.2 SDK 未做任何渲染

只提供适当渲染所需的参数

SDK 0.4中的新渲染后端

延续了关键的渲染特色

Game(App) layer gives层将SDK左、右眼纹理ovrHmd_EndFrame()



SDK渲染与游戏渲染的比较

- 一般工作流程

- `ovrHmd_CreateDistortionMesh`
通过UvS来转换图像
比像素着色器的渲染效率更高
让Oculus能更灵活地修改失真
- `ovrHmd_BeginFrame`
- `ovrHmd_GetEyePoses`
- 基于EyeRenderPose(游戏场景渲染)的立体声渲染
- `ovrHmd_EndFrame`

SDK渲染与游戏渲染的比较

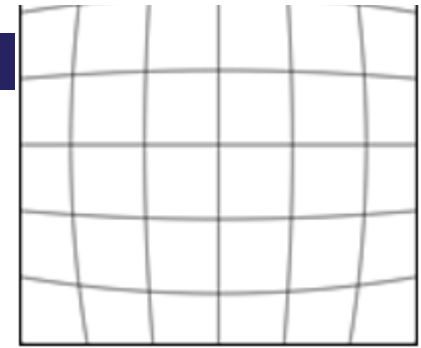
用以下特色在SDK 0.4上完成渲染：

利用色差和时间扭曲来实现桶形畸变

内部延迟测试及动态额预测

Low-低延迟垂直同步（v-sync）和翻转（用direct-to-rift，甚至会更好）

健康与安全警告



SDK集成

- 易于集成：
 - 无需创建着色器和网格
 - 通过设备/系统指针和眼睛纹理
 - 支持OpenGL和 D3D9/10/11
 - 必须为下一帧重新申请渲染状态
- 好处：
 - 与今后的Oculus HW和Features更好地兼容
 - 减少显卡设置错误
 - 支持低延迟驱动显示屏访问，例如前缓冲区渲染等。
 - 支持自动覆盖：延迟的测试，摄像头指南，调试数据，穿透，平台覆盖

支持游戏引擎

Unity 3D/Unreal Engine 3与4使用SDK渲染

在某些情况下，你认为我们的SDK不支持的情况，可能实际是支持的；但我们仍然需要知道！

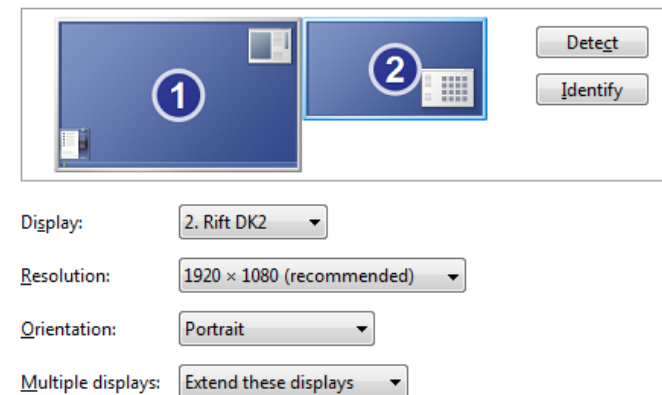
告诉我们。我们会倾听。



扩展模式

- 头戴设备显示为一个OS Display
- 应用程序必须将一个窗口置于Rift监视器上
- 图标和Windows在错误的位置
- Windows合成器处理Present
- 通常有至少一帧延迟
- 如果未完成CPU和GPU同步，则有更多延迟

Change the appearance of your displays

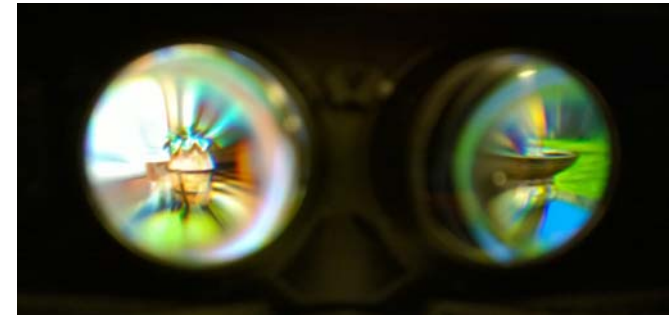


Direct To Rift

输出到Rift，显示未成为桌面的一部分

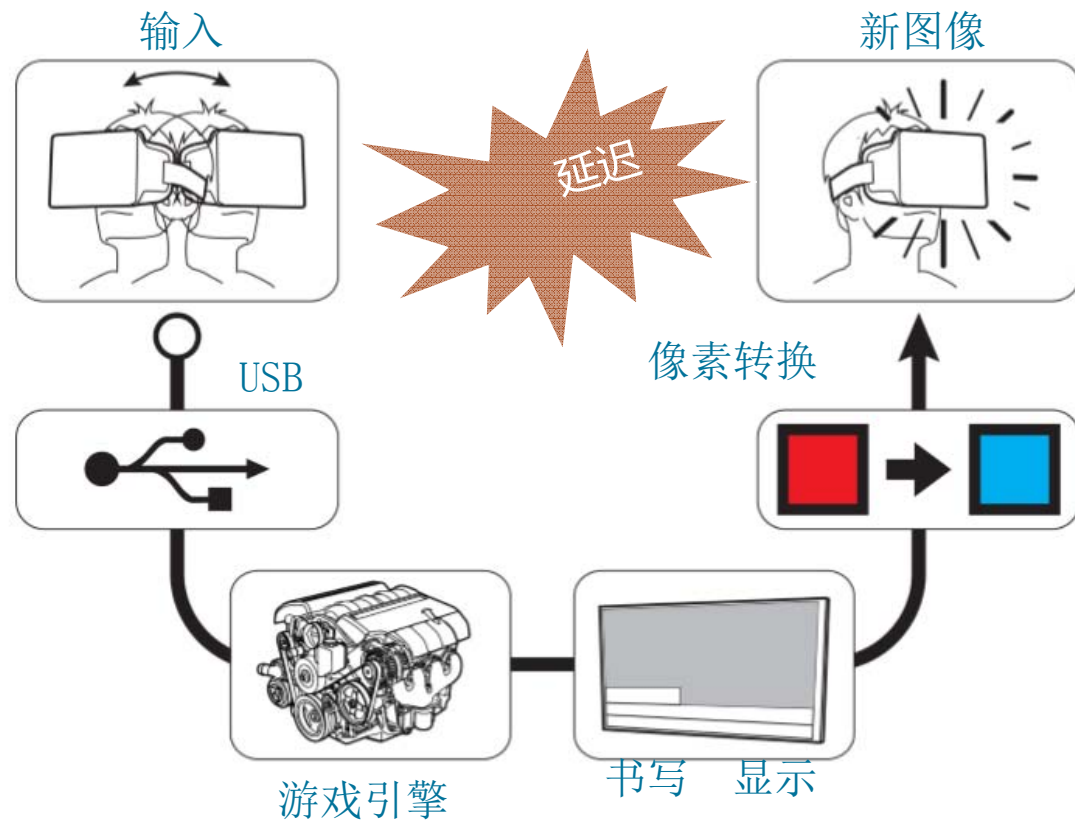
- 头戴设备未被操作系统看到
- 避免跳跃窗口和图标
- 将Rift垂直同步（v-sync）与OS合成器分离
- 避免额外的GPU缓冲，使延迟降到最低
- 使用ovrHmd_AttachToWindow
- 窗口交换链输出被导向Rift

希望直接模式成为较长期的解决方案



5. 延迟

- Motion-to-photon时延
- 多阶段
 - 动作
 - 传感器
 - 处理与合并
 - 渲染
 - Scanout
 - 传输
 - 像素变化时间
 - 像素余辉

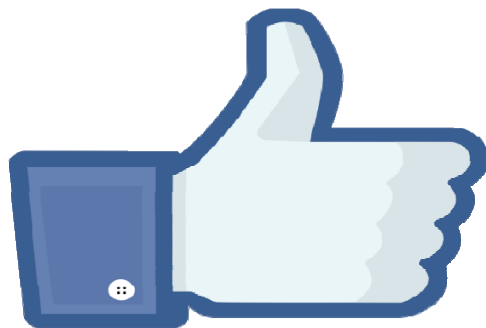


5. 延迟

将延迟保持在低值，是提供良好虚拟现实体验的关键

目标是 < 20毫秒，希望接近5毫秒

我们快要达到了！（请关注我们的最新数据）



渲染延迟 - 时间扭曲

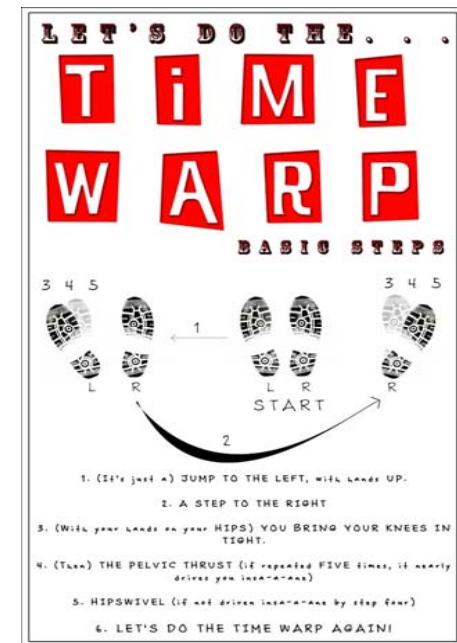
将渲染重新延迟到后面一个时点

与变形同时进行

减少感受到的延迟

负责DK2滚动快门

0.4 SDK处理方向、位置



减少延迟 - 时间扭曲

在帧结束前, 使用传感器是否有其他方式?

时间扭曲 - 预测的渲染 - John 首创

13.3毫秒

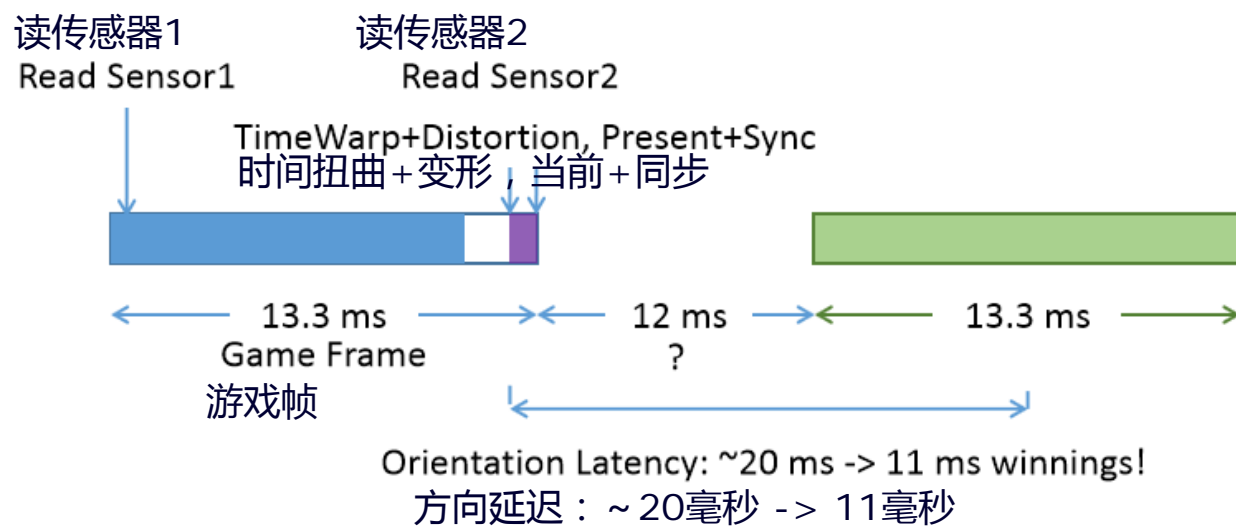


75 FPS

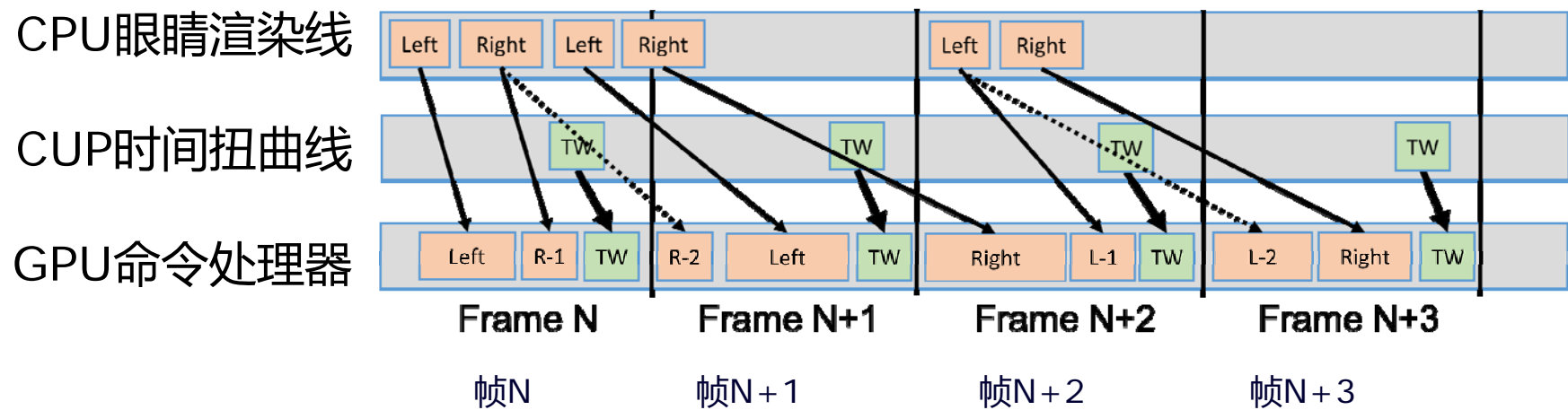
减少延迟 - 时间扭曲

在帧结束前, 使用传感器是否有其他方式?

时间扭曲 - 预测的渲染 - John 首创



减少延迟 - 时间扭曲



6. SDK 路线图

驱动程序的改进与修正

- 多显示器固定到垂直同步相关的延迟和振动
- 减少post-present延迟
- 处理Optimus和多GPU的情况

Linux支持

异步/自适应时间扭曲

- 通过在必要时产生额外的帧，减少振动

时间扭曲层

- 将驾驶舱和覆盖与场景分来进行处理

