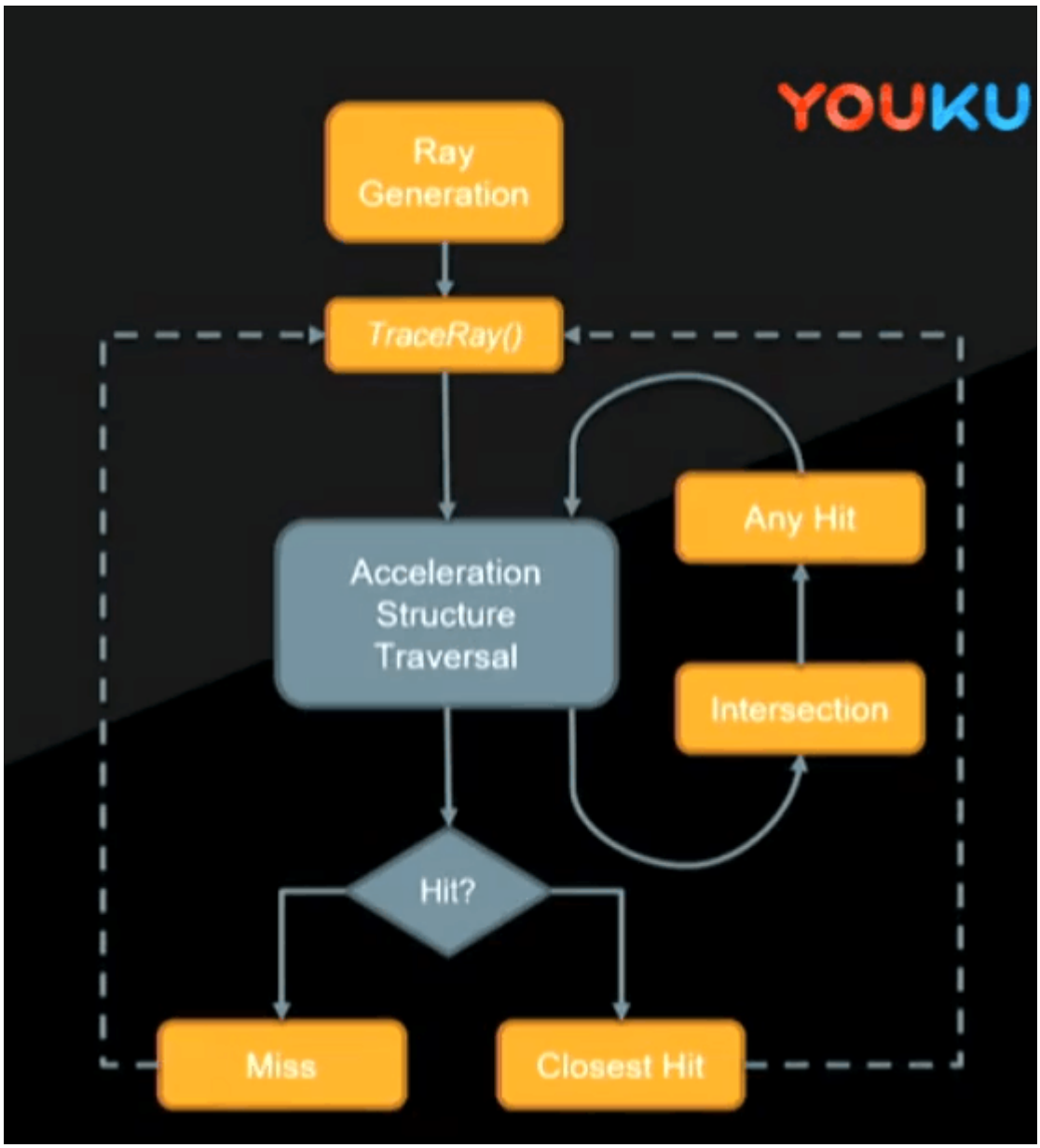
NVIDA real time rending

1080p 24帧 《反射》

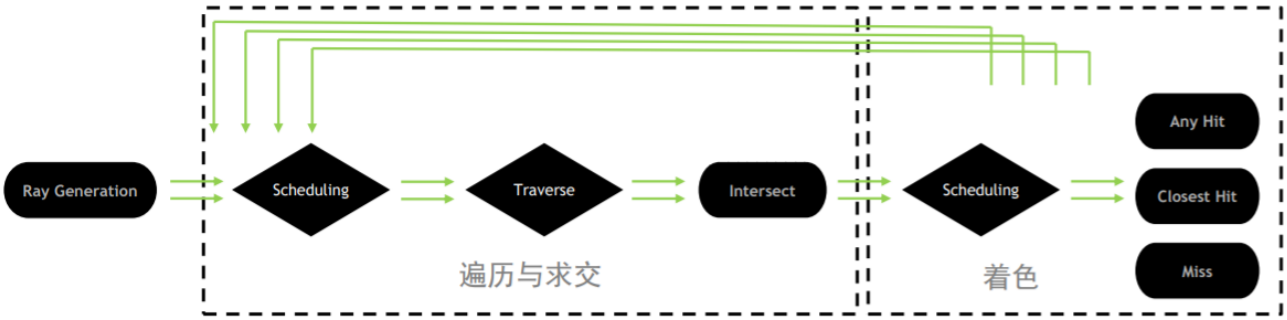
4 turing GPU: 10 Glga rays/sec

结合光栅化和光线追踪？ 混合渲染



payload 结构承载数据 TMIN-TMAX

RTX光线追踪构架



- 混合执行固定功能单元与可编程单元
- 固定功能硬件单元RT Cores: 调度(Scheduling), 遍历(Traverse), 光线-三角形求交
- 可编程单元Shaders: Ray Generation, Intersect, Any Hit, Closest Hit, Miss

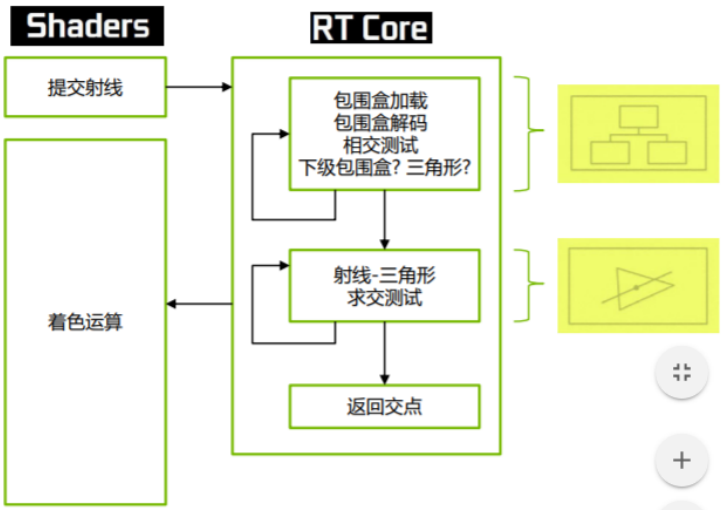
RT Cores

RT Cores提供遍历和求交功能

- 遍历加速结构：自顶向下搜索BVH
- 射线-三角形求交：返回重心坐标和交点距离

调用RT Core

- 光追管线中的Shader提交射线至RT Core输入队列
- Shader可继续执行其它操作
- RT Core返回结果，Shader执行进一步运算



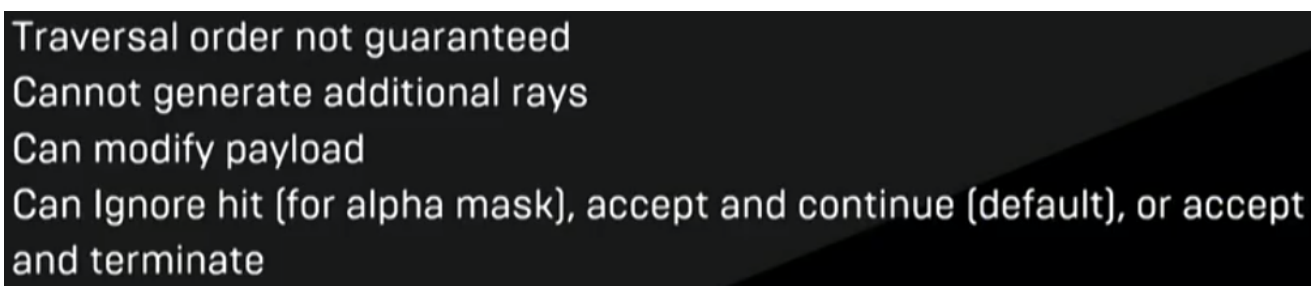
Ray generation

计算简单，栅格化，不需要多线程

Ray Intersection

三角是最基本的形状

Any hit shader



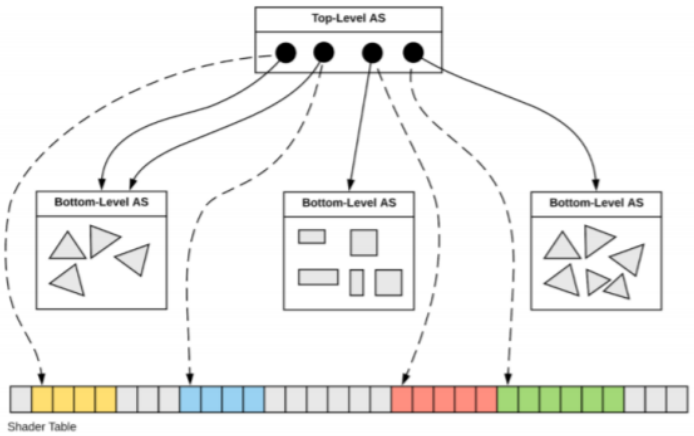
Closest hit and miss shaders

Closest Hit invoked once for closest intersection along the ray
Miss shader invoked if not valid hit is found along the ray
After all 'any hit' shaders complete (if any)
May call TraceRay() (trace recursion)
Can read/write payload

Acceleration

加速结构

- 两级层次结构
  - 底层(bottom-level):基本的几何体信息，包含三角形或包围盒数据
  - 顶层(top-level):指向底层几何体的实例、相应的资源索和坐标变换数据
- 具体格式对应用不可见
- 由应用程序调用API显式构建/更新



Two level Acceleration Structure
- Top level contains instances of bottom level data
- Bottom level built over triangles or AABBs

Support full build and update
- Update enables deformable triangle geometry

Reflections

难题：动态场景栅格化，光线追踪需要处理多次反弹而不只是视角可以看到的

去燥：

Sophisticated Denoising developed by NVIDIA

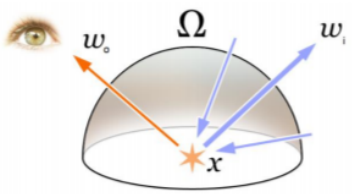
- Glossy reflections reconstructed with adaptive filter guided by roughness
- Relies on custom UE4 TAA pass before applying denoiser
- More details in our talk at 12:45 PM

为什么会有噪点

实时光线追踪

为什么会有噪点？

- 在求解渲染方程时，需使用蒙特卡洛采样：
  - $L(\omega_o) = \int_\Omega L(\omega_i) f(\omega_o, \omega_i) |\omega_i \cdot n| d\omega_i = \frac{1}{N} \sum_{i=0}^{N} L(\omega_i) f(\omega_o, \omega_i) |\omega_i \cdot n| / p(\omega_i)$
- 上式Estimator中的每一项都是半球域上的复杂函数
  - 入射辐亮度(Incoming radiance) L, 可见性 V, BRDF f, 采样密度分布函数 p
- 当前硬件条件下，每像素至多几十个采样（spp/samples per pixel）
- 低采样率导致Estimator的高方差，表现为噪点



Denoiser：

Ray Tracing Denoiser

- 复杂的反射(RT Reflections)
- 面光源软阴影(RT Shadow)
- 环境遮挡(RT AO)

QUESTION:

1.

光线追踪的反射

与SSR对比

- 屏幕空间反射(SSR)的问题
  - 无法反射到屏幕空间以外的几何体
  - 反射的高光不正确
    - 高光依赖于视角，反射光线的视角已经不同于主视角，但依然采样了主视角下的高光颜色

2.

光线追踪的反射

降噪

- 仅对入射辐亮度(Incoming Radiance)进行降噪
  - $L(\omega_o) = \int_\Omega L(\omega_i) f(\omega_o, \omega_i) |\omega_i \cdot n| d\omega_i \approx \underbrace{\int_\Omega L(\omega_i) d\omega_i}_{\text{降噪!}} \underbrace{\int_\Omega f(\omega_o, \omega_i) |\omega_i \cdot n| d\omega_i}_{\text{预积分}}$
- BRDF及高光被预集成，不参与降噪，所以不会糊掉
- 各项异性的交叉双边过滤器
- 使用历史帧数据改善质量及稳定性（类似TAA）

光线追踪的面光源软阴影

降噪

- 仅对可见性项$V(\omega_i)$降噪
  - $L(\omega_o) = \int_\Omega f(\omega_o, \omega_i) L_d(\omega_i) V(\omega_i) |\omega_i \cdot n| d\omega_i \approx \underbrace{\int_\Omega V(\omega_i) d\omega_i}_{\text{降噪!}} \underbrace{\int_\Omega f(\omega_o, \omega_i) L_d(\omega_i) |\omega_i \cdot n| d\omega_i}_{\text{解析逼近}}$
- 可分离交叉双边过滤器(Separated cross bilateral filter)，半径及权重可变
- 辅助信息：交点距离，场景深度、法线、光源大小及方向
- 使用历史帧数据改善质量及稳定性（类似TAA）
- 不同的光源类型使用不同的降噪方案
- 每个光源单独降噪

光线追踪的环境遮挡

降噪

- 可分离交叉双边过滤器
  - 根据交点距离(Hit T)来估算卷积核尺寸
- 开放区域使用较大的卷积核，接触区域（Contact Region）使用较小的尺寸
  - 开放区域中可见性变化较慢，共享范围可以更广
  - 保留接触区域的暗影(Contact Darkening)
- 只需2-4 spp，即可保留接触区域的遮挡细节