总概览：

《使命召唤：无限战争》间接光包含：

Lightmaps

Probe lighting

Lightmaps——大型结构化几何体

Representation

Projection

Light probes——包含下列所有

从光照中分离可见性

可见性的表达

光照插值

光照单元格结构以及存储、访问

生成

烘焙

可见性表达

首先是Lightmaps只是在这里简要介绍了一段时间，我们喜欢它们。它们确实非常简单且快速地查找-对我们而言这是无价的。而且，它们的尺寸与表面的尺寸匹配。这样，光照贴图几何的面积与内存消耗之间就存在线性关系。

为了对方向信号进行编码并允许局部法线变化，我们使用称为AHD编码的AHD编码（代表环境光和高光方向），因此对光照贴图的每个像素都结合了环境光和定向光。我们查看了其他表示形式，Peter-Pike将稍作讨论，但是AHD只是给了我们最大的收益。我们将方向存储在切线空间中，这使我们可以使用半八面体[hemi-octahedral]编码。对于这两种颜色，我们使用BC6h对其进行压缩，尽管与此同时存在一些较小的质量问题，但美工人员可以选择按每个地图禁用压缩。

但是我们知道，Lightmaps并不是灵丹妙药，它们在某些情况下效果很好，而在其他情况下则完全没有效果。诸如小几何形状，灌木丛，具有更大体积结构的事物，Lightmap等性能不佳。在以前的游戏中，我们使用顶点照明或Probe lighting。问题在于顶点照明看起来很棒，但成本很高，因为每个实例都必须有第二个带有定向照明信息的顶点流。另一个上的Probe lighting非常便宜，因为我们只需要为物体存储一个Light Probe，但是看起来并不太好，我们希望找到一种既美观又便宜的解决方案。

现在，这就是典型的探头照明通常看起来的样子。“典型”是指您在一个位置对照明进行采样并将其用于整个对象，通常将其编码为三阶球谐函数，通过法线查找，然后在飞行中通过余弦进行动态卷积。当然，这会给您一些方向性的变化，但总的来说看起来相当平坦且无趣。

我们真正想要的是这样的东西-因此，随着在物体空间变化以及适当的遮挡，这些东西在质量上可以与每个顶点的烘焙光照相媲美。

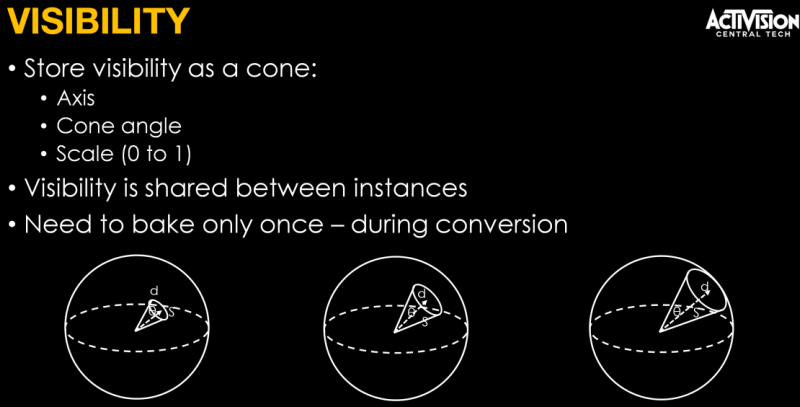
解耦

漫反射表面在无限远的光以及遮挡时的渲染方程：

入辐射度——变化缓慢且平滑

可见性——变化迅速，高频方向成分.

因为当我们执行此操作时，可以以不同的方式对两者进行编码，因此可以以不同的方式对它们进行插值。而且我们可以节省大量内存，因为可见性不再是每个实例都唯一-可以在对象的副本之间共享-照明，即使它是唯一的，它也以较低的空间速率变化，因此 可以以低得多的分辨率存储。在我们的特殊情况下，我们将可见性存储在顶点上，并将照明存储在对象周围散布的一些点中-现在我们将讨论这两个组件。

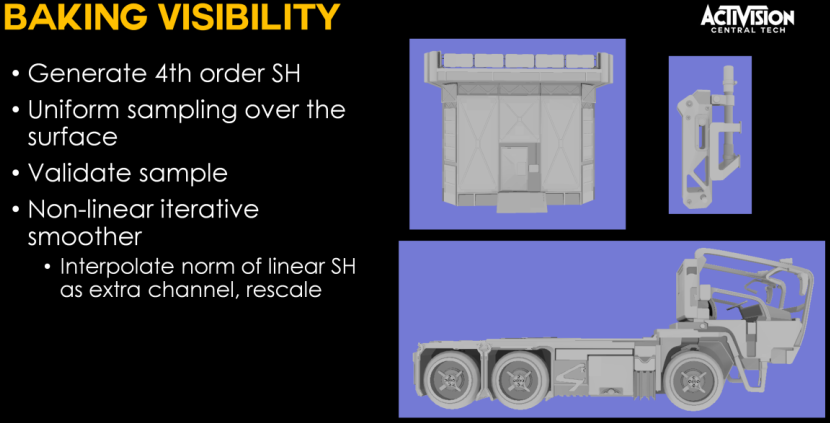


首先，可见性-我们将其存储为缩放的圆锥体-因此，对于网格上的每个顶点，我们都存储其轴-主遮挡方向

然后，我们存储锥角-因此，半球的未被遮挡的部分越宽，锥越宽

然后，我们存储从0到1的缩放系数(Scale)，以具有更大的自由度，以便能够表现出更大的随机可见性，而单个圆锥体可能不是一个很好的表示。因此，当一个点从多个方向遮挡时，我们可以使用一个宽角度并将其按比例缩小，而不是用较小的角度进行补偿。

就像我提到的，可见性在实例之间共享，并且在网格导入期间仅烘焙一次。



为了获得可见性，我们烘焙4阶sh向量。我们对网格区域进行统一采样，对每个此类样本进行验证-检查其是否未被掩埋等，然后从烘焙的样本中得出顶点的值。我们对烘焙的数据进行了几次平滑和平均化处理，以使可见度信号更清晰

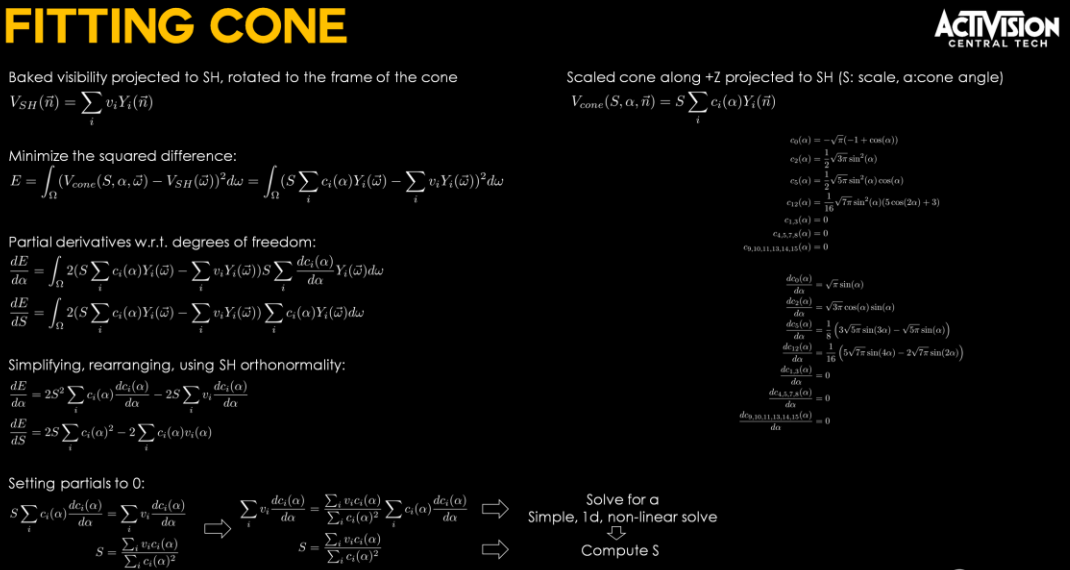
顶点可见性烘焙可能出现的伪像：

1. 画面模糊，某些地方太暗
2. 对物体内部使用较短的射线
3. 刚体组件独立烘焙
4. 透明体没有遮挡，但是他们会产生遮挡信息
5. 骨骼物体没有特别处理，仅仅使烘焙射线稍短一些。

拟合圆锥

最小二乘拟合圆锥

现在我们有了每个顶点的SH可见性，我们将其转换为我前面提到的圆锥表示。对于轴，我们使用来自SH的最佳线性方向。缩放系数和角度的拟合需要非线性拟合过程，但事实证明两件事实际上是分离的，您可以首先求解角度–它仍然是非线性求解，但实际上却很简单，因为这仅仅是 1d –然后将比例计算为视锥上积分的可见性与视锥上自身积分的比率。



这就是调试可视化中的烘焙可见性。如您所见，方向与法线相当一致-这当然是有道理的-主要的非遮挡方向通常是法线方向-但是在遮挡区域中，例如机器人的腋窝，它会偏离很大。角度的余弦看起来相当像AO，但是那是因为它实际上是AO的一种形式-如果不使用余弦项来计算AO，实际上就是这样

Lighting

现在，让我们谈谈照明。我们立即知道的一件事是，整个物体使用单个光探针根本不够–光线可以在整个物体（尤其是大型物体）上发生巨大变化。一种选择是使用辐照度梯度，该梯度在空间上对照明进行一阶泰勒展开，但我们对其质量与成本之比并不满意–最终，它仍然只是一个光探针。因此，对于每个物体，我们实际上对照明使用多个采样位置-我们将它们分散在对象周围，其数量取决于对象的大小-然后从整个对象的这些点内插照明

我们会自动找出那些采样位置。 我们首先弄清楚我们需要多少–这只是基于物体大小的启发式方法。然后，我们对物体进行均匀采样并执行k均值聚类，以将这些采样分组为N个聚类。经过几次迭代后，最终聚类的中心变为采样位置。

给定一组采样点，我们可以得出插值模式。为此，我们计算采样点集的协方差矩阵，并提取其特征值和特征向量。 这些的结构决定了我们如何执行插值-特征值的相对大小决定了插值模式，特征向量形成了发生插值的局部空间。

这就是我们的意思。如果整个物体只有一个探针，那很简单–整个物体仅使用探针。

如果我们有更多的特征值，但是只有一个大的特征值，则意味着采样位置成一条线。可以将对象上的每个点投影到该直线上，并通过在它们之间进行插值来使用两个最接近的探针来获取照明。在这里，我们只有两个探针，但是如果有更多探针并且仍然形成一条线，它们还将使用1d插值模式。

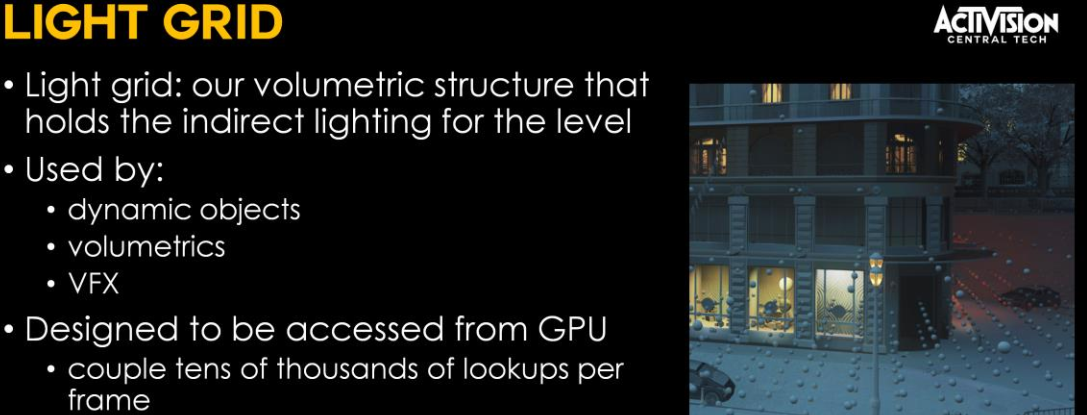
如果存在两个大特征值，则采样点将形成一个平面。因此，我们将所有物体投影到该平面上，对采样点集进行三角测量，对象上的每个点都使用三个探针，它们是其投射到的三角形的顶点。

如果所有三个特征值都很大，则探针将仅形成3d云（我们将其四面体化），并且对象上的每个点都将使用4个探针（其所在的tet的顶点）来获取照明。

因此，总而言之，我们通常生成给定维数的Delauney网格，并将网格的每个顶点分类为所生成的单纯形之一。我们离线进行此操作，对于网格的每个顶点，我们将存储给定顶点使用的探针的索引以及插值权重。即每个顶点额外存储8字节数据(索引+权重)，该数据在实例之间共享。此外，蒙皮物体可以绕开所有物体，并将采样位置明确附加到特定骨骼。

从34页至46页全部都是关于Light Probe和Visibility合成的，需要对球谐有更深入的理解。

Light Grid



我们知道我们不希望艺术家手动放置探针。我们先前依靠的手动系统放置了包含感兴趣区域的体积，因此我们决定重新使用它们

但是我们不能只在其中均匀地放置探针-因为过渡空间的体积可能会很大。因此，我们确定了探针的不均匀分布和该集的四面体化-就像几年前Robert[Cupisz12]在GDC上谈到的解决方案一样。Tets很不错，因为您只有4个值可在其中进行插值，并且结果是C0连续的。

从50页到54页全部都是关于Light Grid。

Light Grid Visibility

从55页到61页解决Light Grid可见性问题

Baking

从62页到84页讨论关于烘焙的问题，这里仅仅是把球谐函数投影到AHD上的问题

球谐函数

从85页到110页讨论如何解决球谐函数导致的环问题。