摘要

在本文中，我们将八面体作为参数化方案进行研究，以表示实时渲染应用程序中的环境图。我们讨论了两种投影方案，并显示了两个方便的展开映射，以将柏拉图式实体填充为二次或矩形纹理。 我们进行分析，并考虑交互式应用程序的渲染性能以及表示质量。 我们将进一步讨论受益于此参数化的应用程序和场景。

1 介绍

环境映射是在渲染中广泛使用的已建立的众所周知的技术。 通常，从空间的一个点看到的整个环境都投影到一个参数化表面上，并存储在一个或多个易于访问的纹理中。 在交互式渲染中，作为将远处物体反射到任意形状的几何上的有效方法，以及通常用于存储方向数据的方法，同样重要。 它在捕获环境照明的照明算法中起着至关重要的作用，并且已经发现了更多的应用，例如全向阴影映射。

本文的结构如下：首先，我们简要概述环境映射技术和相关工作。在第3节中，我们讨论了用于环境图的新八面体表示形式，并描述了必要的转换，然后在第4节中讨论了实现该技术时出现的问题。在第5节中，我们在第6节中简要讨论了我们的结果。

2 之前工作

Blinn和Newell [1]引入了第一个环境图，他们将反射向量转换为球面坐标以访问二维纹理。这种方法在极性区域和可见的接缝处会发生纹理变形。威廉姆斯[12]和霍夫曼[9]引入了球形环境图，将正交视图存储到纹理中的理想镜球上。该技术是第一个获得硬件支持的技术，但是，该方法在观察方向上表现出很强的非均匀采样和奇异性。

Greene [3]引入了立方图，该图映射了当今最流行的技术。它既不会遭受严重的欠采样，也不会遭受奇异之处。它在创建和查找时都兼顾了速度和易用性。最重要的是，它自然可以捕获现实世界以及综合场景。

因为立方体贴图由六个纹理组成（必须在动态场景中渲染），所以Heidrich和Seidel [5]开发了双抛物面贴图，将环境投射到两个镜面抛物面上。这种方法不会共享球形环境映射所固有的缺点，也不会出现奇异或严重的欠采样情况。此类地图的创建相当复杂，而且并非没有限制，因为其下的投影不允许像立方图那样直接使用相机（不存在此类镜头）进行直接创建，并且它们也不基于线性透视图投影。也就是说，需要高几何细分以减少近似误差，或者需要更多涉及的渲染算法[2]。

3 八面体环境图

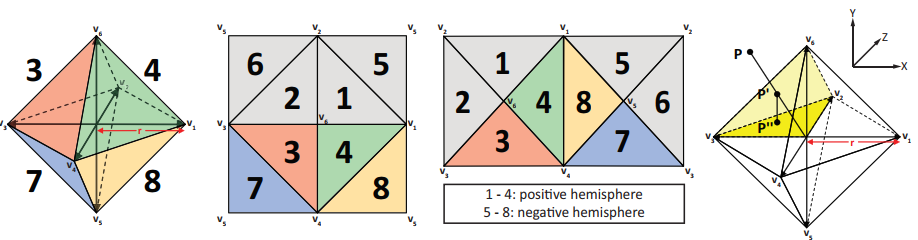
Praun 等人[10]已经使用八面体作为球形域的一种可能的参数化方案，他利用球形域到2D纹理的一对一映射来存储球形几何图像。我们将八面体用作实时渲染应用程序的环境图，包括使用图形硬件生成和查找八面体环境图（OEM）（有关示例应用程序，请参见图7）。

八面体的每一侧代表一个方向的八分圆，我们描述了两个投影以将方向映射到侧面上的点。此外，八面体的侧面在二维纹理中的排列可能不同（请参见图1）。

在下文中，我们假设所有坐标都相对于八面体的坐标系给出，如图1（右）所示。当然，在渲染过程中设置适当的转换很容易实现。

3.1 平面投影

我们分析的第一个投影方案类似于球形投影.如果,则点位于半径为的球面上,则类似地,如果,则点位于具有顶点的八面体表面上.因此,我们得出一个投影方案,我们将其表示为点的平面投影（另请参见图1）：



**图1**：我们将环境投影在八面体上，该八面体可以展开并包装成一个二次方或矩形纹理.我们展示了如何直接渲染到这些纹理中以及如何有效地查找八面体环境贴图.对于投影,先将点投影到八面体面上,然后将其正交投影到平面上.

为了将八面体展开为二次纹理，正半球上的点即被正交投影到XZ平面上(图1).通过分裂与相邻的所有边缘来展开下半球.因此,可以通过以下方式获得投影点:

其中是符号函数.

我们的第二种展开方法将八面体映射为纵横比为2:1的矩形纹理,如图1所示.首先应用等式1,然后将点正交投影在平面上,并从2个半球投影坐标通过以下转换:

根据布局映射和必须映射到相应的纹理坐标以进行访问和渲染.

3.2 透视投影

对于立方体贴图，可以将透视投影视为获得八面体环境贴图的另一种方法。八面体的每个侧面可以渲染一次场景，每个场景都有自己的投影矩阵。在这种情况下，与立方体映射相反，平截头体具有四面体形状，即，我们具有三角形的视口，图形硬件本身不支持剪切，因此必须在对象或对象中手动执行 图像空间。

使用这些投影查找八面体图的方法类似于立方体图：首先必须确定查找向量的八分圆，然后是投影变换和八面体布局的映射（如上所述）计算。

为了完整起见，我们检查了此参数化，但是与立方体贴图相比，没有任何预期的优势（8个投影而不是6个投影，外加附加裁剪）。因此，在本文的其余部分中，我们将重点放在平面参数化上，该参数更简单，因此对某些应用程序有利。

4 实现

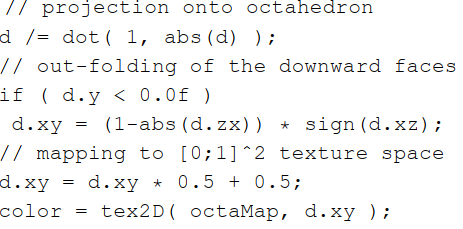
4.1 直接渲染八面体贴图

平面投影不能用矩阵机制来表示，因为不能相应地表示绝对值（参见等式1），但是可以很容易地与可编程图形硬件一起使用。此外，重叠八角形边界的三角形无法正确处理：GPU上的变换和投影是按顶点执行的，每当三角形相交于八角形以上时，扫描线转换或光栅化阶段（假设线性三角形边缘）都会产生错误的结果（如图所示） 2）。请注意，这有点类似于抛物面贴图的投影误差。类似地，我们可以通过使用精细细分的几何形状来保持较低的误差，但是为了产生正确的结果，需要付出额外的努力，下面对此进行了介绍。

线性插值在每个八分圆内都是正确的，即，我们可以通过在八分圆边界处分割三角形来解决该问题。我们可以利用几何着色器：在XY-，YZ-，ZX平面上将三角形拆分为较小的三角形（最多27个），这样每个输出三角形仅驻留在一个八分圆中。裁剪后，将计算投影坐标，并根据布局（二次或矩形）应用仿射变换。请注意，对于基于点的渲染技术，由于点精灵的尺寸较小，如果误差可以忽略，则可能不需要强制使用特殊八角形边界处理。

4.2 纹理查询

**未经过滤的纹理查找** 可以类似于前述的投影和布局变换来计算用于给定方向向量的八面体图中的纹理坐标。与二次布局一起使用时，未过滤的查找非常有效。可以通过以下几条指令（着色器伪代码）来计算八面体坐标系中给定方向向量d的纹理坐标：



**过滤后的纹理查找** 对于Mip映射查找，我们需要考虑两个方面。首先，在确定纹理查找所需的Mip级别时，该决定基于被变换为纹理空间的屏幕空间像素单元所跨越的四边形区域的扩展。它是根据纹理坐标及其偏导数计算的[8]。请注意，对于按像素计算的反射，通常根据表面曲率或光泽度手动计算mip级别。我们用dX和dY表示纹理空间中的偏导数。然后，可以通过以下方式得出mip映射级别：

但是，在八面体贴图的情况下，四边形区域可能会与由于展开而分开的两个相邻八分音符相交。对于二次布局，当四边形位于负半球时会发生这种情况。图3（中心）描绘了二次布局的可能情况，但矩形布局也发生了类似的配置。

因此，当一个单元格跨越纹理空间中两个不相交的区域时，我们需要进行特殊处理。 幸运的是，八面体的对称性使我们能够反射下半球到上半球的像素单元的坐标，从而产生一个连续的四边形，然后将其用于计算正确的mip级别（图3，右）。在实践中，我们计算原始单元和镜像单元的Mip级别，并使用这两个值中的最小值。

对于矩形布局，两个半球都映射到连续区域。因此，当使用纹理数组（DirectX 10支持）时-每个半球使用一个纹理-然后将自动正确处理过滤，也用于每个顶点的纹理坐标计算。 如果不使用纹理阵列，则必须对上述二次布局进行类似的处理。

过滤后的纹理查找的第二个方面是，由于展开了对纹理的环绕访问，因此无法正确生成双线性插值的颜色值（这适用于抛物线形，并且也适用于立方贴图）.最好通过引入“安全边界”来解决，即通过复制像素（如图4所示），以便可以使用用于双线性插值和mipmapping的样本。

请注意，最大正确表示的Mip级别取决于安全边界的宽度。我们还将八面体图与求和的面积表一起使用，以渲染光泽反射[6]。在这种情况下，我们使用由BRDF的最大光泽度确定的安全边界宽度（图7显示了一个示例）。请注意，为了进行精确的BRDF建模，需要多个SAT查询，这些查询适用于所有球形参数化和（预）过滤的环境映射。

5 结果

5.1 采样

为了评估八面体图的采样质量，我们计算每个像素覆盖的立体角（相对于球体在像素总数上的立体角）。图5显示了八面体参数化与立方体贴图的比较。最高采样率接近八面体顶点，但在整个曲面上都处于可接受的范围内。正如预期的那样，平面投影比透视投影产生更均衡的采样。整体质量与立方体贴图相当。

5.2 直接渲染

我们进一步测量了足够复杂的场景（图8）的渲染性能，创建了三个近似相等像素数的环境图。我们直接渲染512×512二次布局（QL）OEM，具有两个362×362纹理的矩形布局（RL）OEM和六个209×209纹理的立方体贴图。时序在表1中列出。我们已经在4.1节中描述的几何着色器阶段中通过显式裁剪实现了平面投影和透视投影。我们表示这种方法，它最多可以生成81个顶点（严格分割）。如3.2节所述，透视OEM的裁剪也可以通过丢弃不落入三角形视口的像素在图像空间中完成。我们称此方法为剪贴蒙版。

对于矩形布局，我们给出了平面投影的结果，并且仅研究了两种不那么精确的裁剪技术：在几何着色器中，其性能在很大程度上取决于输出到后续管线阶段的顶点数量的裁剪被证明是我们实现过程中的瓶颈。利用两个半球都映射到连续区域这一事实，由于与XZ平面相交，我们仅拆分了映射到两个纹理的三角形。我们表示这种方法，它输出两个最多7个顶点的三角带，XZ分割（图6，XZ分割）。

第二种方法称为边缘条，它通过XZ分割在两个半空间（正和负y轴）中分别创建的顶点进行处理。 这些顶点形成凸多边形，其边缘被其余的分割平面所分割（图6，边缘分割）。 然后，根据获得的点直接创建两个三角形带（图6，三角形带）。 请注意，尽管这种天真剥离保留了两个多边形边界上投影的正确性，但不能保证正确的插值，例如 三角形条带内的纹理坐标的集合。 当然，这是不正确的，但是比错误的原始形状可以容忍。 此方法总共输出最多15个顶点。

5.3 应用和优势

尽管直接渲染OEM带来了许多困难，但它们提供了两个独特的功能。首先，OEM能够在整个球形域上生成总面积表（SAT）。在GPU上创建SAT时，例如 采用Scheuermann等人[6]的方法。输入和输出参数化可以独立选择。这是因为生成需要多次渲染遍历，并且第一个遍历可以包含重新参数化，即，也可以将易于渲染的立方体或抛物线贴图用作输入。此外，查找OEM厂商是使方向数据可用于所有着色器阶段的有效方法。 对于顶点或几何着色器，不支持立方体贴图，第4.2节中介绍的OEM查找比对手工计算的立方体或抛物面贴图的查找更有效率。

当要存储多个（半）球形域时，八面体参数化还提供紧凑的存储。（半）球形阴影贴图用于即时光能传递方法[7]或场景辐射的稀疏采样[4]。 代替抛物线形参数化会浪费多达20％的纹理空间，OEM可以打包而不会浪费纹理空间到纹理地图集中。

作为OEM，经纬度环境贴图允许将球形域参数化为矩形纹理，因此，它们还可以计算SAT，提供紧凑的存储以及在顶点和几何着色器中查找。但是，采样情况更糟，并且在极地区域表现出两个奇点，查找需要三角函数，因此操作昂贵，渲染到纬度-经度图也需要裁剪和拆分。