

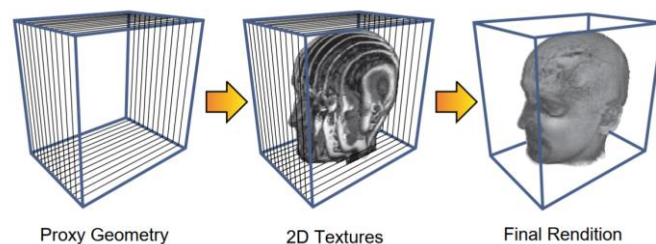
基于切片的先进体照明技术

本文详细介绍了基于切片的体照明技术，该技术基于早期 GPU 纹理方法，是将性能与质量平衡到极致的体照明方法之一。

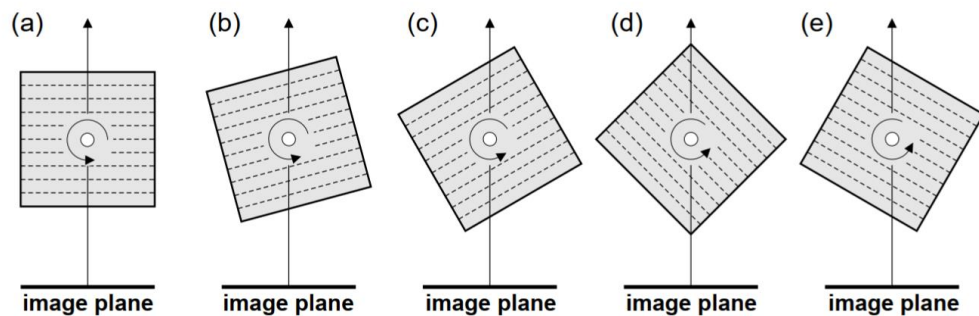
一、 基于切片的体渲染技术

书籍[1]有该方法的详细过程。

最简单的想法是轴对齐的切片，使用 alpha 混合：

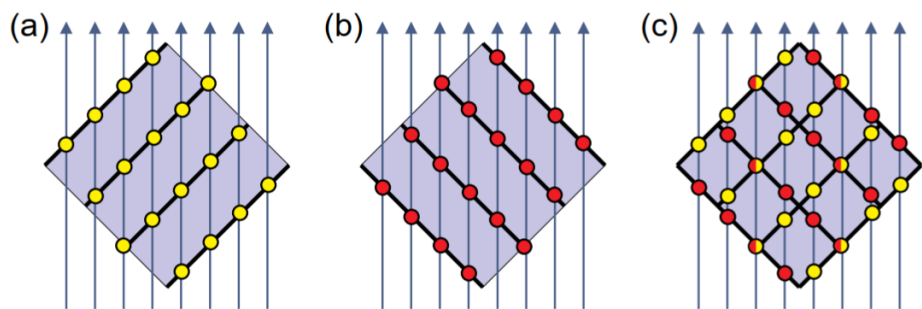


注意，为了保证轴的方向与成像平面尽量一致，需要对三个轴存储三份切片，以保证实时交互时随时切换轴：

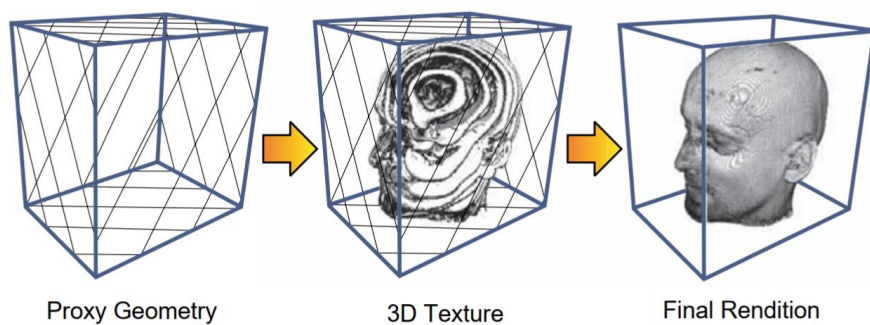


注意，合成时，OpenGL 里的纹理合成机制需要合成轴按顺序，例如当合成选项为 `GL_ONE_MINUS_SRC_ALPHA`，表示的合成方式是先渲染后面的，然后放前面的。还可以设置 alpha 是否对颜色先进行预乘等选项。注意如果先渲染前面的切片，再渲染后面的切片，则需要其他参数设置（后面半角切片技术就是先渲染后面的切片）。

但是这样会带来 switching 伪影，如上图(d)，从(c)到(e)会因为轴突然改变，导致出现采样的点突变，带来渲染效果的突变，如下图所示：



因此，开发了基于视角对其的切片技术，生成一个代理体数据：



计算平面盒交点的一种方法可以表示为三个步骤的序列。

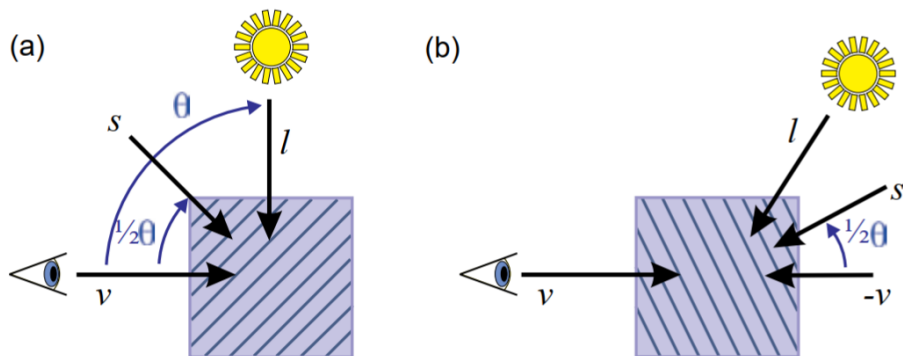
1. 计算切片平面和表示边界框边缘的直线之间的交点。
2. 消除重复和无效的交点。如果平面与所在直线相交，但交点不在边上，则表示该交点是无效点。与边界框的角顶点重合的重复点将合并在一起。
3. 对剩余的交点进行排序，形成一个闭合多边形。

之后，多边形的顶点值可以存储其实际位置，这样中间的插值就是自动完成的。代理集合体就可以访问到原体素的所有值了。

后面的混合方式与之前完全一样。

二、 半角切片技术

当光源与视角方向夹角小于 90 度时，切片方向为夹角的一半，否则就是光源与视角反方向夹角的一半，如下图所示：



注意上图(a)和(b)的合成方向是相反的，因为只有这样才能同时计算光衰减和体渲染合成。渲染需要两个 buffer，一个光 buffer，一个视角合成结果 buffer。光 buffer 存储每层光衰减后信息，需要进行不断更新。

光 buffer 先与当前视角合成结果相互作用（相乘），然后再更新到被下一层 slice 衰减后的值。之后再与下一层视角合成相互作用，再继续更新。

三、 多重散射与间接光照

该技术主要基于[3]，当然论文[3]还提出了其他技术，但我们只关注多重散射部分。Full-全局照明、后向散射和体发射属性并不在我们的考虑范畴。

$$I_{eye} = I_B * T_e(0) + \int_0^{eye} T_e(s) * g(s) * f_s(s) * I_l(s) ds$$

$$I_l(s) = I_l(0) * \exp\left(-\int_s^{light} \tau(x) dx\right)$$

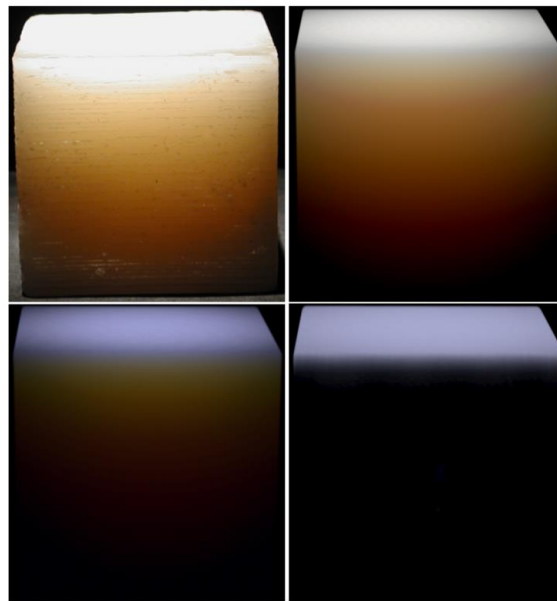
$$I_{eye} = I_0 * T_e(0) + \int_0^{eye} T_e(s) * C(s) * I_l(s) ds$$

$$C(s) = g(s) * ((1 - S(s)) + f_s(s) S(s))$$

$$I_l(s) = I_l(0) * \exp\left(-\int_s^{light} \tau(x) dx\right) + I_l(0) * \exp\left(-\int_s^{light} \tau_i(x) dx\right) \text{Blur}(\theta)$$

其中， $g(s)$ 可以表示为反射比例， $S(s)$ 可以表示为表面反射比例，可以用梯度模来控制。间接照明的净效果实际上是光线通过体的扩散。光在体中的传播距离比只考虑直接衰减时更远。半透明意味着由于散射效应，光在穿过介质时会变模糊。我们可以简单地模糊某个邻域中的光线并允许其在光线方向上较少的衰减来近似这种效果。

下图显示了我们的模型如何捕获半透明效果。左上图是用聚焦手电筒从上方照射蜡块的照片。右上图是使用我们的模型（具有白色反射色和去饱和橙色传输色（1-间接衰减））进行的体绘制。请注意，与单独使用直接衰减（体阴影）相比，光穿透到材质中的深度要深得多。右下图显示了仅考虑直接衰减的光传输效果。左下图显示了仅将反射颜色更改为淡蓝色的效果（左下图像具有明亮的蓝色反射色和与右上图像相同的传输色）。还要注意，由于间接衰减项使蓝色比红色或绿色衰减得稍微多，因此从白色到橙色再到黑色的明显色调转移效果（也就是说，设置反射蓝色更多以后，蓝色衰减的就不会那么明显了，所以渲染效果的色调转移效果变弱了）。



指定消光系数不如直接指定 alpha 项更容易理解：

$$\alpha = \exp(-\tau(x)).$$

我们的模型还向传递函数添加了一个间接衰减项。这个项是彩色的，这意味着它描述了 R、G 和 B 颜色分量中每一个的光的间接衰减。

$$\alpha_i = \exp(-\tau_i(x))$$

用 $1-\alpha_i$ 来表示传输色更好，这样调节传输色就能看到显示的结果。

我们的体绘制管线计算光在体中的传输，与眼睛的光积累同步。该方法使用体阴影计算的半角切片，其中切片轴位于光和视图方向之间的一半，或者位于光和倒置视图方向之间，具体取决于两者的点积符号。

实现我们的模型总共需要三个图像缓冲区。除了为观察者积累光的缓冲区（通常是帧缓冲区）外，还为光方向（当前和下一个）的光衰减保留了两个缓冲区。

在第一个过程中，从观察者的角度渲染切片。在这一步使用反射颜色和阿尔法的

依赖纹理读取来计算转换函数。在硬件片段着色阶段，反射颜色乘以 1 减去先前在当前光缓冲区中该切片位置计算的**间接和直接光衰减之和**。然后使用 alpha 值将此颜色混合到观察者缓冲区中。

在第二个过程中，从光源的角度将切片渲染到下一个光源缓冲区中，以计算下一次迭代的照明。保留了两个光缓冲区，以适应间接衰减所需的模糊操作。我们在片段着色阶段显式计算混合，而不是使用标准的 OpenGL 混合操作来混合切片。使用渲染到纹理 OpenGL 扩展，在第一轮中为观察者对当前光缓冲区采样一次，在第二轮中为光采样多次。然而，下一个光缓冲区仅在第二遍中渲染。这种关系在第二轮后发生变化，因此下一个缓冲区成为当前缓冲区，反之亦然。我们称这种方法为乒乓球混合。

[1] Engel K, Hadwiger M, Kniss J M, et al. Real-time volume graphics[M]//ACM Siggraph 2004 Course Notes. 2004: 29-es.

[2] Kniss J, Kindlmann G, Hansen C. Multidimensional transfer functions for interactive volume rendering[J]. IEEE Transactions on visualization and computer graphics, 2002, 8(3): 270-285.

[3] Kniss J, Premoze S, Hansen C, et al. Interactive translucent volume rendering and procedural modeling[C]. IEEE Visualization, 2002. VIS 2002. IEEE, 2002: 109-116.