

唐无忌 07.05 – 07.12

本周主要整理了光栅化的内容，包括走样反走样、卷积滤波等，之后学习了着色，包括 Blinn-Phong 反射模型，三种着色模型，以及纹理映射的内容，双线性插值，MipMap 算法，之后又看了一点几何的内容。

Rasterization（光栅化）：

视口变换：

计算机图形学中，计算机的屏幕（screen）被抽象地定义为一个二维数组，数组中的每一个元素称为像素（pixel），数组的大小（即所含像素的个数）称为分辨率（resolution）。

在将空间中的各物体的世界坐标转换到了在相机空间下的相机坐标，并把相机空间中的任意一个截锥内的物体投影到了 xy 平面上的标准正方形（相机的接收区域） $[-1,1]^2$ 中，接下来就是把接收区域内的点变换到计算机屏幕的对应位置上，这个变换就叫视口变换。

光栅化：

在把物体变换到了屏幕空间中，下一步就是把它们“绘制”到屏幕上。光栅化（rasterization）指找出屏幕空间中的物体所覆盖的像素、并确定这些像素应该呈现什么颜色，从而实现将物体绘制到屏幕上这一操作。

研究三角形的光栅化本质上是确定这样一个函数：输入一个三角形（由三个顶点表示），输出能够尽可能准确地显示这个三角形的一系列像素的位置及它们的颜色

采样：

简单来说，就是查询一个函数在某一点处的值，通过采样，可以将一个连续的函数离散化。光栅化就是一个采样的典型例子：把“连续”的图形在每个像素处采样，得到图形在采样处的值，并以此为基础生成以像素为单元的最终图像。对边界处的点不做处理的话，就会产生锯齿现象，需要做反走样处理。

反走样：可以先对三角形进行模糊化处理，再采样。

现在，我们把图像（已被光栅化）当做一个有界的、二维的、离散的信号，而把屏幕空间中的物体（还未被光栅化）当做有界的、二维的、连续的信号。于是，对图像的处理本质上属于信号处理的一部分。我们的目标是分析与处理信号，为此，我们需要发明与运用各种各样的方法，Fourier 变换是信号处理领域最普适、最有力的数学工具之一。

卷积定理：

若在时域中把两个信号 f 和 g 的卷积进行 Fourier 变换，在频域中得到的是 f 和 g 的 Fourier 变换的乘积；若把 f 和 g 的乘积进行 Fourier 变换，则得到 f 和 g 的 Fourier 变换的卷积。

有重要结论：（1）在时域中对 f 进行以 T 为周期的采样后再进行 Fourier 变换，在频域中等价于将 f 的频谱以 $1/T$ 为周期进行平移复制；（2）采样频率越高，频谱周期越大、显得越稀疏；采样频率越低，频谱周期越小、显得越密集。

走样的原因：当我们采样频率过低时，频谱周期较小，原信号频谱的相邻两个复制谱之间可能会有重叠，即在同一个频率上可能有来自不止一个复制谱的分量。

反走样的方法主要有两种：

（1）增加 f_s 即增加分辨率。这是最直接的方法，但相应地要生产更好的屏幕，花销较大；（2）减小 f_{max} ，即把原信号的频谱中高频的部分“删去”。在信号处理中，常常出现把信号中频率位于某个区间内的分量删除的操作，这种操作就叫滤波（filtering）。

滤波：

高通滤波保留了图像的轮廓，即那些变化剧烈的部分，也就是图像作为信号的高频分量；低通滤波“舍弃”了图像的轮廓，保留了变化缓慢的部分，也就是图形的低频分量，达成的效果是使得图片变模糊。这与直觉是一致的： f 变化愈剧烈，所包含的高频分量愈丰富；变化愈缓慢，所包含的低频分量愈丰富。

于是，我们知道了：模糊 \approx 低通滤波 \approx 某种卷积。而计算机实际上无法进行这个卷积操作：卷积是在屏幕空间里进行的， x 和 y 都是连续量，计算机只能处理离散的量。但其实根本无需求出 f 在每一个点的值，只需求出 f 在屏幕上每个像素的中心处的值，因为光栅化只要求获得这些点的值，这些值即为这个像素最终呈现的颜色。因此，做卷积（滤波）和光栅化（采样）两步看起来像是合并成了一步：在采样点求卷积值。

MSAA：

把一个像素拆成很多个“子像素”（尽管实际上并没有），在每个子像素中心采样（即计算该位置的 *inside*），所有子像素的值做平均后的结果即为这个像素最终的值。它并没有增加屏幕的分辨率，只不过在每个像素内部做了更多的采样，用这些采样的结果的平均近似该区域三角形面积的占比。相比于不做反走样，代价自然是花费了更多时间。

Shading（着色）:

着色:

所谓“着色”，狭义上讲，指确定光线打在一个物体上呈现出的效果；广义上讲，指将一种材质应用到一个物体上。

Blinn-Phong 反射模型: $L=L_a+L_d+L_s$

其中， L_s : 高光（spectacular highlight），源于镜面反射； L_d : 漫反射（diffuse reflection）光； L_a : 环境光照（ambient lighting），也是某种漫反射。

着色频率（shading frequency）: 指的是应用着色模型计算颜色的频率。

- (1) **平坦着色**，着色频率是面，对每个三角形面仅作一次着色，整个三角形都显示为这个颜色。
- (2) **Gouraud 着色**，着色频率是顶点，在每个三角形面的每个顶点着色，内部的颜色通过对三个顶点的颜色插值得求。
- (3) **Phong 着色**，着色频率是像素，对每个像素都计算一次着色。

重心坐标（barycentric coordinates）:

是一种三角形的插值算法，它能够根据三角形每个顶点的值（数或向量）求出内部的值，并且使得整个三角形内该值平滑变化。

图形管线（graphics pipeline）:

是指从构建三维场景到最终渲染出一张图呈现到屏幕上中间经历的全部操作过程，包括顶点处理、三角形处理、光栅化、片断处理和帧缓存操作。

着色过程发生在顶点处理和片断处理中，例如如果是 Gouraud 着色，则发生在顶点处理中；如果是 Phong 着色，则是片断处理。在处理过程中，对于每个顶点或片断，将运行一段（通用的）程序，其决定了对于这个顶点或片断应如何着色。这段程序被称为**着色器（shader）**。

纹理映射:

我们可以将一张二维的图片贴到一个三维模型上，使之呈现不同的外观。这样的图片就叫做纹理图片（texture map/image），或纹理（texture）。纹理没有改变三维模型的形状。纹理贴图所在的二维空间被称为纹理空间（texture space），从世界空间到纹理空间的映射被称为纹理映射（texture mapping）。纹理贴图上的每一个像素被称为纹素（texel）。

纹理过小可采用双线性插值、双三次插值，纹理过大可用多级渐远纹理。

双线性插值:

基本思想是取离纹理坐标最近的四个纹素，将它们的颜色进行某种加权平均后赋给该坐标。

多级渐远纹理(Mipmap):

简单来说就是一系列的纹理图像，后一个纹理图像是前一个的二分之一。
当距观察者的距离超过一定的阈值，就使用不同的多级渐远纹理，即最适合物体的距离的那个，由于距离远，解析度不高也不会被注意到。