

各位好，我是dz1933026 王国畅，今天介绍一下catmull在texture mapping方面的工作。

翻页

catmull是这次的图灵奖得主，获奖是由于他在3D和计算机生成图像（CGI）方面作出了巨大贡献。catmull的贡献主要包括曲面细分技术，z-buffering（有许多同学会介绍这个），Texture Mapping（纹理贴图），图像合成，以及他提出了Reyes渲染架构并设计实现了RenderMan渲染引擎，他的贡献不止于此，我在这里只列举了部分。他的研究被广泛应用在电影，游戏中CG动画的设计，例如皮克斯和迪斯尼的各类动画电影（他本身是皮克斯和迪斯尼的工作室成员和总裁，因为这些工作在获得图灵奖之前也被多次授予了奥斯卡的动画科技奖）。

我们今天着重介绍他在texture mapping（纹理贴图）方面作出的贡献。

翻页

这里我们可以举个例子来说明纹理贴图的大致效果。右侧的图片中，左上方是不包含纹理贴图，由计算机直接生成的坦克，它的每个平面都是光滑的并且颜色统一，显然可以看出这是一个坦克，但是看起来并不真实。经过了纹理贴图的处理它会变成右下角的样子。两者的区别在于右下角的坦克看起来是有纹理的，例如装甲上本来是平整一片，经过处理后添加了表面的装甲接缝和文字，这时坦克看起来就真实了许多。

纹理就是这种处于表面的细节，通过添加许多凹凸不平的纹理，3D图形就会更加真实。早期计算机生成图像最为人诟病的就是曲面过于光滑导致的失真，添加纹理的做法可以很好的改善这一点。这些纹理的生成主要有两种方式，一种是计算纹理的曲面，添加到原有的3D图形中，一种是直接对曲面上绘制纹理。前者就好比在坦克模型的表面雕刻纹理，后者好比在坦克模型的表面贴一层贴纸，因此这项技术用中文翻译成纹理贴图十分形象。在这两种方式中，纹理贴图也是更加被广泛使用的手段。

翻页

以这页上的三张图片为例，通过给这个圆球添加纹理，我们可以得到若干不同纹理效果的圆球，通过纹理贴图，我们只需要处理5张图片的投影，而如果去计算小球上每一处纹理的曲面，则需要在原有的3D图形中添加众多的曲面，使得图形更为复杂。说完了纹理贴图是个啥，纹理贴图很好和我们需要纹理贴图，下面再来介绍一下纹理贴图具体是怎么工作的。

翻页

首先我们形式化的定义一下纹理和纹理贴图，纹理指的是一个多维的图像被投影到一个多维空间。纹理贴图指的是一个纹理首先被投影到一个3D空间的物体表面，被投影到被观测的屏幕上。这里的定义取自皮克斯的一篇survey，作者叫Paul S. Heckbert，同时也是cmu图形学课程推荐的文章，我的水平难以理解其中精妙，感兴趣的同学可以自己去看这个定义，我这里就取个图片给大家一个感性认识。纹理就是像左侧这个砖块一样的表面，纹理映射就是把这个砖块贴纸，先贴到中间的一个3D曲面上，然后把这个贴了纹理的曲面显示在显示屏上。中间涉及两次变换的过程，笼统的称作map，所以这项技术叫texture mapping。

翻页

我们最终只需要在显示屏上得到添加了纹理的图像即可，这个目标可以通过很多算法实现，主要包括三种，第一种是Screen Scanning，它遍历屏幕的每一个像素，通过纹理和3D表面来计算这个像素应该显示什么，我们把显示器类如画布，这种算法就像一个画家观察模特以后在画布上画画。第二种是texture scanning，它遍历纹理的每一个像素，并计算纹理最终在显示屏上的投影，这个过程跟人的直觉较为符合，可以类比成有一张平整的纹理贴纸，我们先将它贴到物体表面，然后拍一张照片，照片就是显示屏的图像。可以看出前两种一种是从显示屏出发得到图像，一种是从纹理出发得到图像，都是完成一次2D图像的变换，第三种算法则是把一次2D图像的变换拆分成两段1D的变换。第一个过程沿着x方向进行纹理投影，第二个过程沿着y方向进行纹理投影。第三种算法可以类比成我们在绘图软件里对纹理进行编辑，先横向调整到跟3D图形表面一致，再竖向调整到跟3D图形表面一致。

翻页

第三种two pass算法是由catmull提出的，这种算法的提出是为了解决前两个算法难以解决的问题。首先第一个算法看似可靠，但实际上是一个反向的变换，遍历屏幕的像素来计算纹理的位置复杂度很高，而且需要纹理任何位置都可以被随时读取，因为当我们从左到右从上到下遍历屏幕的像素时，其对应像素在纹理平面上被遍历的次序并不是这样从左到右从上到下的。第二种算法看似和纹理贴图的定义很接近，但是通过遍历纹理像素的方式计算投影会导致画面空洞和overlaps，例如纹理是一个平面纹路，3D图形是一个球面，球面是不能展开成平面的，因此通过遍历纹理的方式计算纹理到球面的投影，就会出现空洞。我们可以把贴纸平整的贴到正方体表面，却不能平整的贴到球体表面。第三种方法的特殊之处在于把2D纹理的变换换成了两轮1D的线形变换，线形变换易于计算，开销小，线形变换也可以在各类曲面表面随意拉伸变换，同时也并非反向变换，因此计算复杂度上和成像效果上都优于前两种方式。

翻页

two pass算法虽然对投射和仿射变换都有良好的效果，但在复杂的曲面上计算复杂度和成像效果同样会打折扣。纹理贴图算法能良好工作的另一个重要条件同样来自catmull的研究，即曲面细分技术，catmull设计了曲面细分的算法，将曲面细分为若干个patches，并通过这个算法给出的曲面细分可行范围界定了two pass算法的适用范围。

翻页

由于对纹理的扫描是线性的，且通过曲面细分的方式简化了曲面，因此计算简单，成像效果特别好。另一方面由于是1D投影，因此图像的处理可以是线性的，从而可以通过流水线进行简单处理，极大方便了电影等行业中工业化的成像过程。也许这项研究本身也是为了便于当年皮克斯的产出。

翻页

谢谢各位，我的pre结束了。