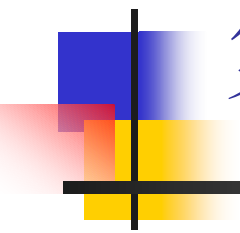
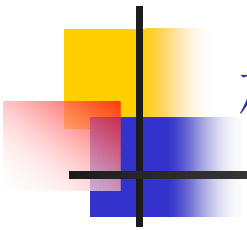


第九章　加速算法

浙江大学CAD&CG国家重点实验室

秦学英

2004年9月



核心内容

- 实时绘制的三大追求
  - 更多的每秒帧数
  - 更高的解像度
  - 更多的物体与更具真实感的场景
- 本章的核心内容是基于空间数据结构（spatial data structure)，在此基础上讨论剔除算法。LOD以及基于点的绘制也简要地涉及。

9.1 空间数据结构

数据结构用于加速查询一个几何物体是否覆盖，其应用包括：

- 剔除算法
- 用于相交测试及光线跟踪
- 用于碰撞检测

通常这种结构是层次的

- 层次包括体 Bounding Volume Hierarchies(BVHs)
- 各种BSP树
- 八叉树

9.1.1 层次包围体 BVH

包围体是包含了一组物体的体

- 球体
- 轴对齐包围盒(Axis-aligned Bounding Boxes (AABBs))
- 有向包围盒(Oriented Bounding Boxes (OBBS))
- 离散有向多面体（K-DOP）
- 包围体只是一种组织结构，对场景本身没有贡献，不予绘制

BVH的结构 树形结构

图9.1

树的节点数

- 树的儿子： k
- 树的高度：$h=\lceil \log_k n \rceil$
- 树的节点总数：

$$n = k^0 + k^1 + \cdots + k^n - \frac{k^{n+1}-1}{k-1}$$

测试计算

以光线与几何物体的交为例：从根结点开始

- 如果光线为了此结点的BV相交，如果其有子结点，检查其所有子结点；反之，返回上层
- 如果光线为此结点的BV无交；光线将不与其中的任何物相交，结束，返回上层
- 此结点为几何物体本身，与物体求交
  - 是否叶结点
    - 若是
    - 若不是

动态物体

- 当一个包含在一个BV中的物体移动了，检查此物体是否包含在父结点中
  - 如果是，这个BVH仍然有效
  - 如果没有，便要更新父结点，直到所有的更新都完成
- 以递归方式增大其父结点的包围体，一直到包围体可以包含这个子物体为止。
- 无论如何，一个紧致的包围体都是必要的。

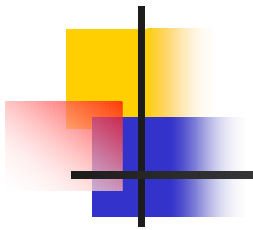
9.1.2 BSP树

Binary Space Partitioning Trees

- 利用一个平面来剖分空间，然后将几何物体归入这两个子空间
  - 轴对齐BSP树(Axis-aligned)：平面是平行于轴的
  - 多边形对齐BSP树(polygon-aligned)：平面为景物中的任意平面
- 递归进行，选择终止条件
- 树要平衡，才可以获得好的效率

轴对齐BSP树

- 建立方法：
  - 首先整个物景被包围在一个Axis-aligned Bounding Box(AABB)中，然后递归地进行剖分，考虑任意级的递归其中的一个盒子去选择盒子的一个轴，然后生成其垂直面，并用来剖分空间。
  - 不同的方法选用了不同的比例来剖分空间：恰好一半或可变比例
  - 对于与分割平面相交的物体，一种处理方法是将其作为整个父结点正或负，另一种方式是将此物体分成两部分，各属一个子结点。
  - 这个过程递归至，树的层次上足够多，或者尺寸已达到用户定义的阈值。



- 例：地形 一个建筑物 一个房间

图9.2

其它策略

- 1.轮流选择x、y、z轴作为剖分轴向(k-d Trees)
- 2.剖分包围盒具有最大尺寸的边，并设图片比例以使BSP树更平衡

多边形对齐的BSP树(Polygon-aligned BSP tree)

- 在场景中选择一个多边形平面来剖分空间，所有与此面有交的多边形都为此面剖分为二，好的BSP树建立标准树是平衡的。
- 剖分策略
  - 最少交叉：先随机选择几个多边形，与其它多边形相交最少多边形被选来剖分空间，测试表明，对1000个多边形的物景。

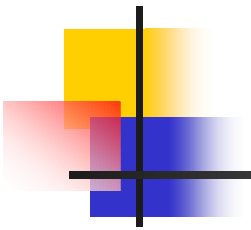


图9.3

9.1.3 八叉树Octrees

图9.4

- 八叉树
- 松弛八叉树
  - 修改：每个包围盒的尺寸是可以稍微大一点，这样使被剖分的物体更少一些
  - 便于移动物体的表达
  - 由于个物体的表达在一个八叉树结点中，其删除是复杂的

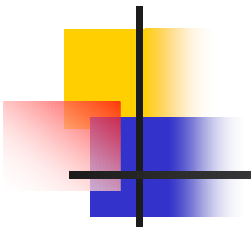


图9.5 松驰八叉树

9.1.4 场景图(Scene Graph)

- 场景图是高级树形结构，以表达场景中的各种信息，除了物体等几何信息，还有纹理，坐标变换，LOD(levels-of-detail)、绘制状态（材料属性等）、光源以其它信息。用深度优先方式遍的树来绘制整个场景

  例：几何体的相互引用

  VRML/Open Inventer

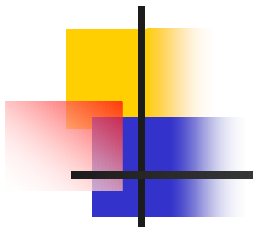


图9.6

- 运动是实时绘制的动力

9.2 剔除技术(Culling Techniques)

- 剔除意即从一个群体中删除出去。
  - 可见性剔除(Visibility Culling)
  - 碰撞检测
  - 物理计算
  - AI
- 在这一节，仅对与绘制相关的裁剪技术进行讨论
  - 背面剔除(Backface Culling)
  - 视域剔除(View Frustum Culling)
  - 遮挡剔除(Occlusion Culling)
  - 严格可见物体集EVC(Exact Visible set)
  - 潜在可见物体集PVS(Potentially Visible set)

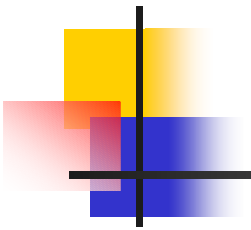


图9.7 不同的多边形剔除技术

9.3 背面/群聚背面剔除
9.3.1 背面剔除 Backface culling

- 背向视点的面被剔除
- 剔除率50%
- 投影多边形的法向计算

$$n = (v_1 - v_0) \times (v_2 - v_0)$$

n为$(0,0,\alpha)$或$(0,0,-\alpha), \alpha > 0$

- 如果-z轴指屏幕，则前者为多边形正面，后者为背面

- 图9.8

9.3.2 群聚背的剔除(Clustered Backface Culling)

- 是决定一组多边形是否应该被绘制的算法。其中的一个重要概念是法向锥(normal cone)

- 图 9.9

9.4 层次视锥剔除(Hierarchical View Frustum Culling)

- 流程：
  - 对从根结点起递归检查BVH的与视锥的交
  - 如果无交，BVH中的所有子结点或其中包含的多边形与视锥都无交，结束处理
  - 如果有交，检查其子结点与视锥的交；如果其子结点为叶结点，则将其送入绘制管道
  - 如果被包含无视椎中，则BVH中所有叶结点被送入绘制管道

层次视锥剔除

图9.11

- BVH
- BSP
- Octree

9.5 入口裁剪(Portal Culling)

- 基本思想
  - 在室内场景中，建筑物墙面通常充当大的遮挡物，通过每个入口进行视锥裁剪。当遍历入口时，就减小视锥，使得与入口尽可能紧密贴合。

图9.12

对场景进行预处理

- 将场景分为一系列单元，它们通常对属于建筑物中的房间或走廊
- 连接邻接房间的门和窗户称为入口
- 存贮方式
  - 墙面可以存贮在一个与单元关联的数据结构中
  - 用手工方式来创建这个邻接图
- 算法框图
- 其它应用：通过变换视点来生成镜面反射，如迷室生成

9.6 细节剔除

- 基本思想：当视点处于运动中时，场景中的微小细节对绘制图像的贡献非常小甚至没有。

9.7 遮挡剔除(Occlusion Culling Algorithms)

图 9.13

图 9.14

基本思想

- 最大限度地剔除掉被遮挡的物体
- 分类：
  - 基于点的遮挡剔除
  - 基本单元的遮挡剔除
- 3种算法空间：
  - 图像空间(image space)
  - 物体空间(object space)
  - 射线空间(ray space)

目标

- 一种遮挡剔除算法的伪代码
- G是将要绘制几何物体的集合
- O_R是遮挡表示
- P是可以和O_R合并的潜在遮挡集合

图 9.16

9.7.1 遮挡地平线

- 是一种在图像空间中基于地平线的遮挡剔除算法，算法是基于点的

图 9.21

遮挡地平线

- 在预处理步骤中，在场景的x,y平面上创建一个四叉树
- 两个重要操作：
  - 如何对物体进行遮挡测试
  - 如何将物体的遮挡指数增加到地平线上

9.7.2 遮挡物与是锥扩张

- 使用基于点的遮挡算法来生成基于单元的可见性，通过对场景中的所有遮挡物收缩一个给定距离来扩展点可见性的有效性

图9.22

9.7.3 轴遮挡剔除

图9.23

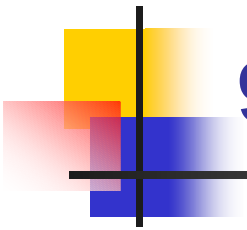
图9.24

9.7.4 硬件遮挡查询

- 使用硬件，通过比较Z值，确定一组多边新的可见性
- 算法的一种扩展为，返回通过可见性测试的像素个数
- 使用层次形式来使用遮挡查询

9.7.5 层次Z缓冲算法

- 算法用八叉树来维护场景，将画面的Z缓冲器作为图像金字塔，也称为Z金字塔
- 算法作用于图像空间



9.7.5 HOM算法

- HOM算法即层次遮挡图(Hierarchical Occlusion Map, HOM)算法
- 算法利用了图形硬件的优势，还可以处理动态情形

HOM算法

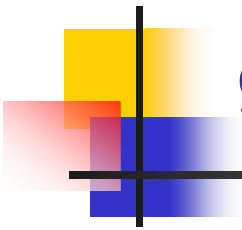
图9.28 深度估计缓冲器的示意图

9.7.6 射线空间遮挡算法

- 基本思想：计算在对偶空间（射线空间）中的可见性信息
- 该算法在二维上是精确的，在2.5上是保守的

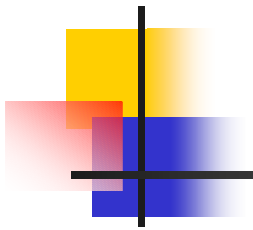
9.8 LOD

- 细节层次(Level of Detail)的基本思想是使用物体的一种简单形式来表达物体
- LOD算法的三个主要部分
  - 生成
  - 选择
  - 切换

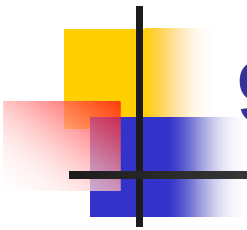


9.8.1 LOD切换

- 离散几何LOD
- 混合LOD
- Alpha LOD

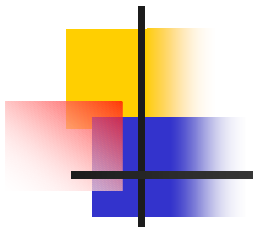


- CLOD和几何变形LOD



9.8.2 LOD选取

- 基于距离的LOD选取
- 基于面积的LOD选取



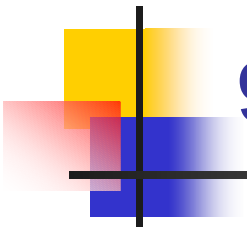
- 基于滞后的LOD选取

图936灰色区域是LOD技术中的滞后区域

- 其它：物体的重要程度、运动、滞后、焦点

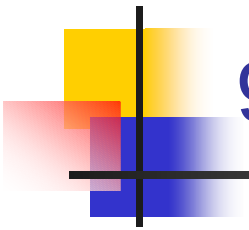
9.8.3 时间临界LOD绘制

- 在特定时间内，完成相应的任务
  - 启发式算法



9.9 大型模型的绘制

- 通常是几种模型的组合



9.10 点绘制