谁遮住了我？

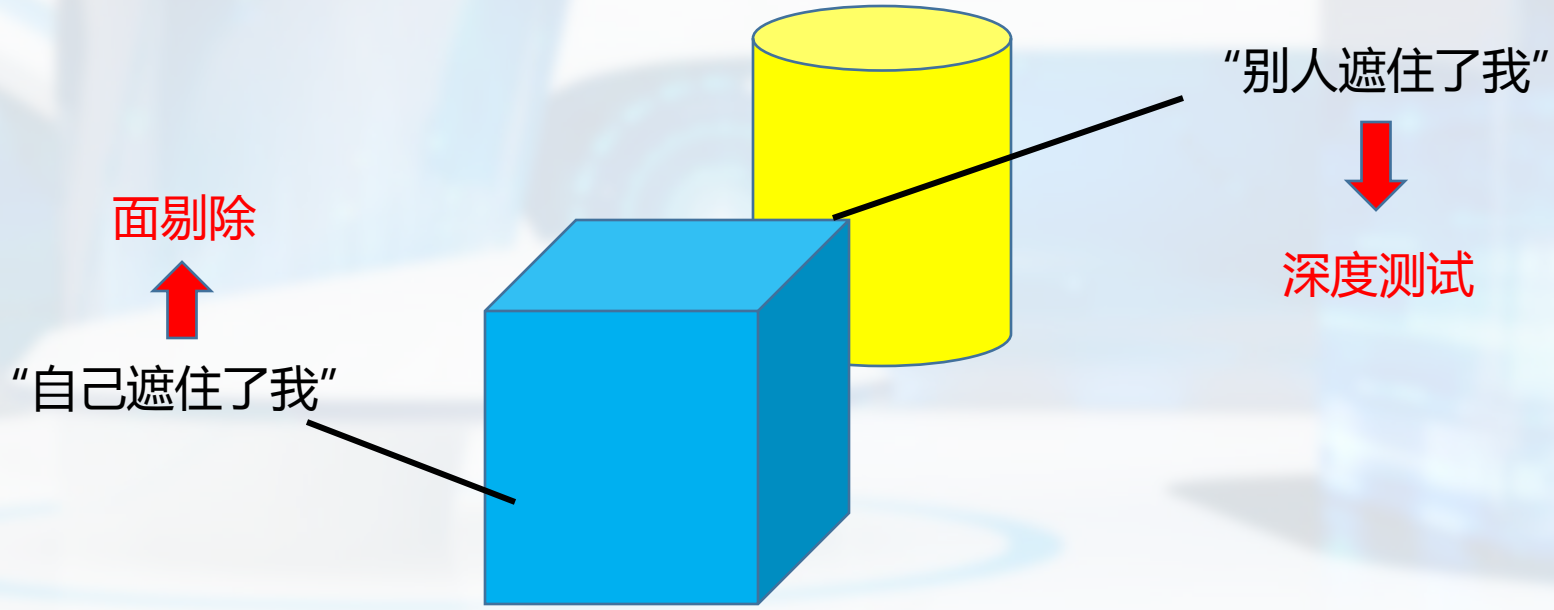
华中科技大学软件学院　万琳

提纲

1. 消隐的概念
2. 面剔除
3. 深度测试
4. OpenGL中的消隐

1 消隐的概念

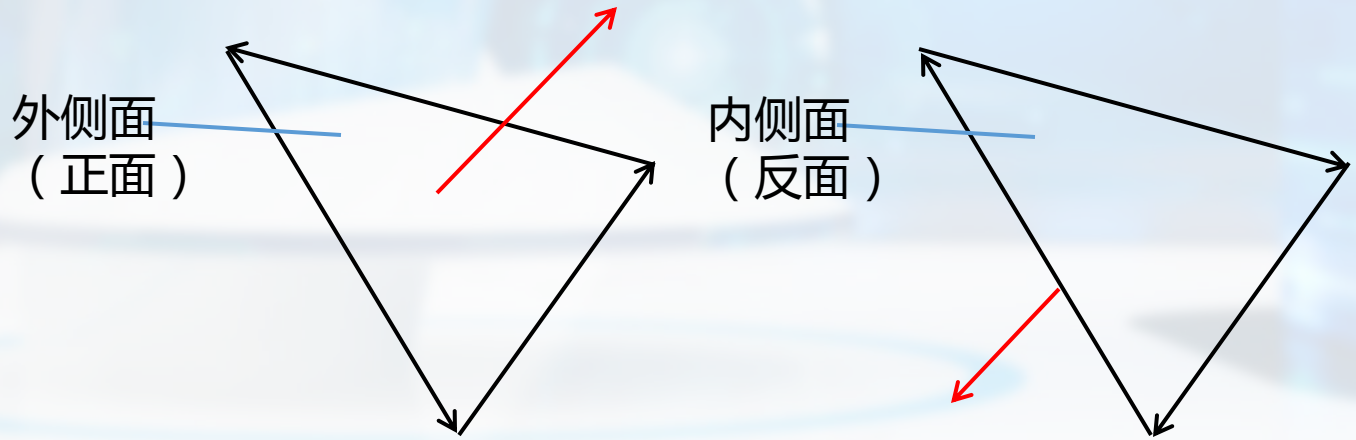
消隐：决定场景中哪些物体的表面是可见的，哪些是被遮挡不可见的



2 面剔除

◆ 面剔除分析

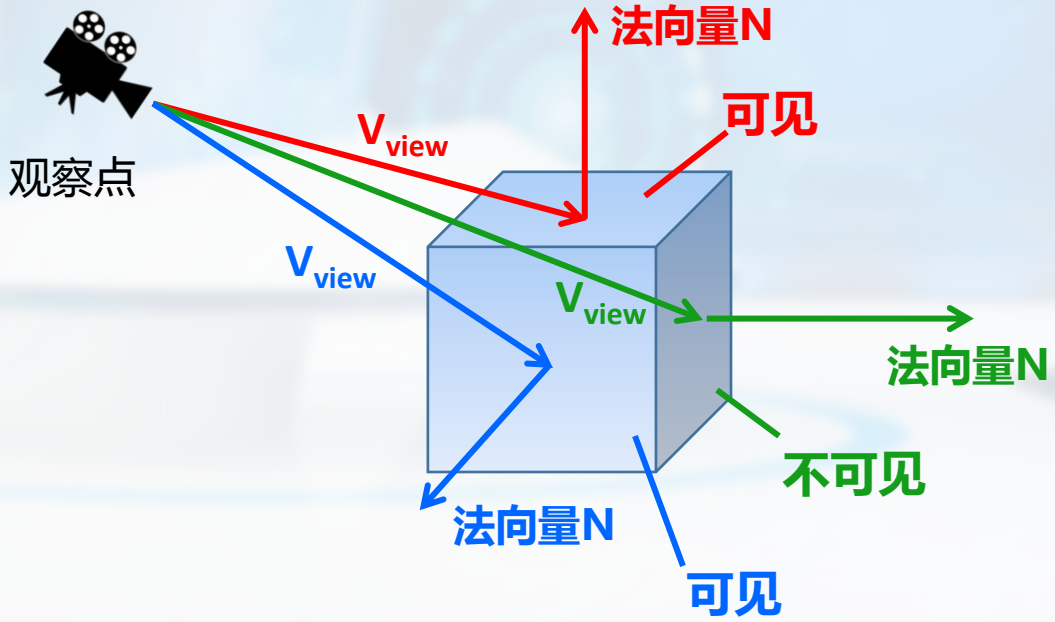
正反面的定义



2 面剔除

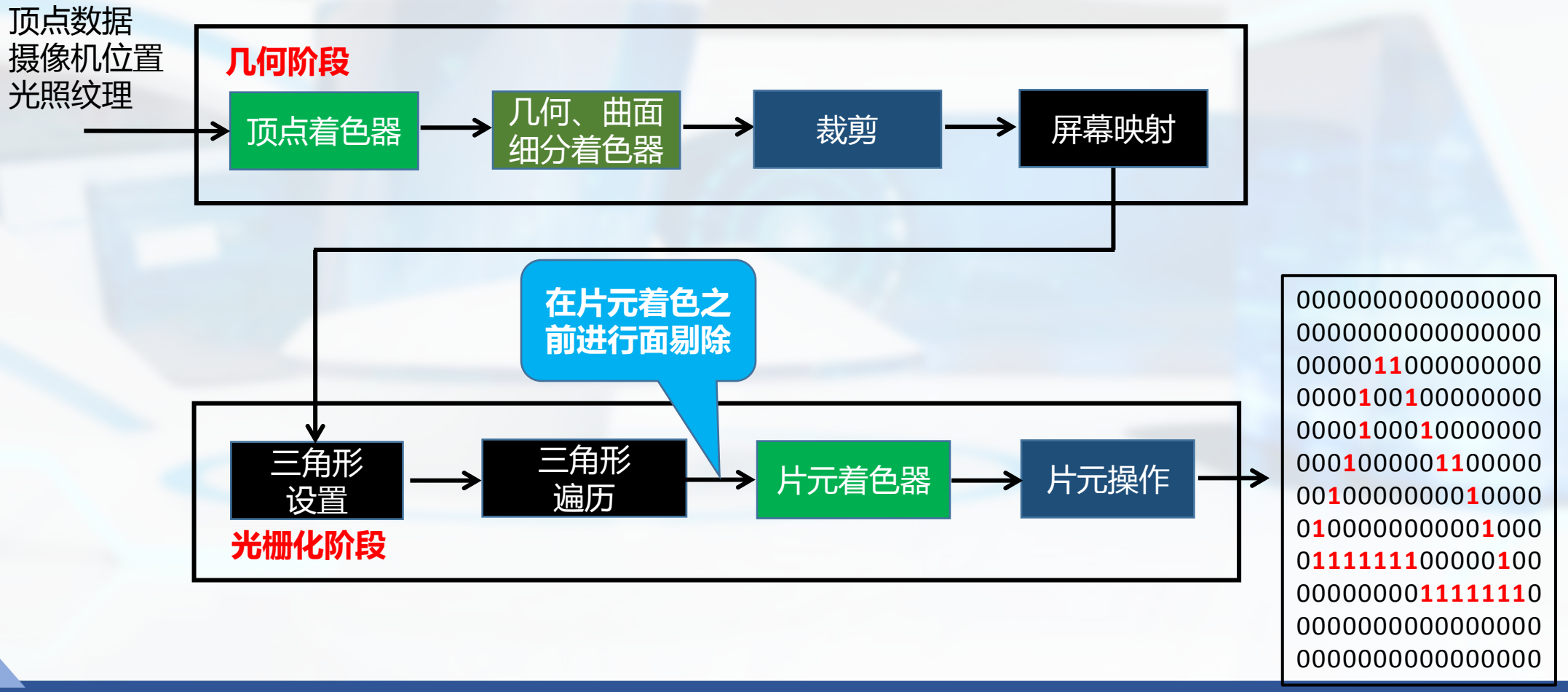
◆ 后向面判别

如果$V_{view} \cdot N > 0$则该多边形为后向面，后向面是不可见的



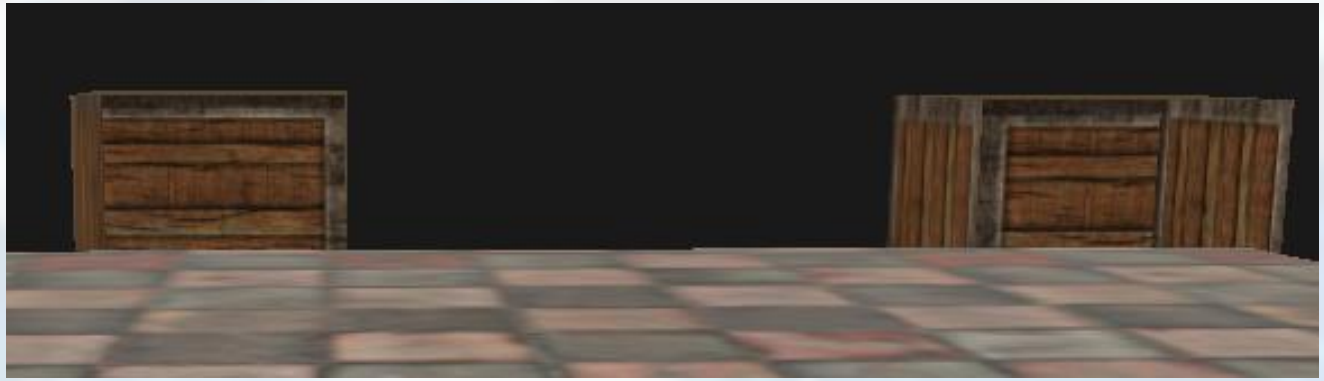
2 面剔除

◆ 什么时候做面剔除？



3 深度测试

- 深度测试的必要性



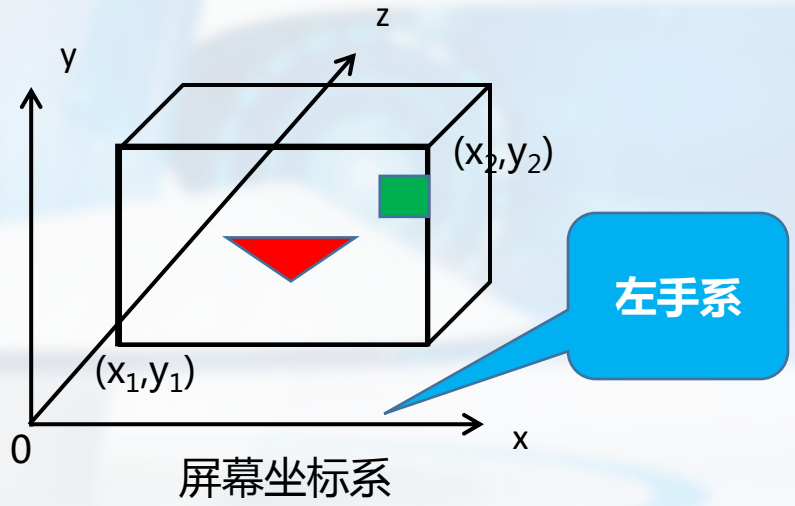
3 深度测试

- 深度缓存器算法
- 深度排序算法

3 深度测试

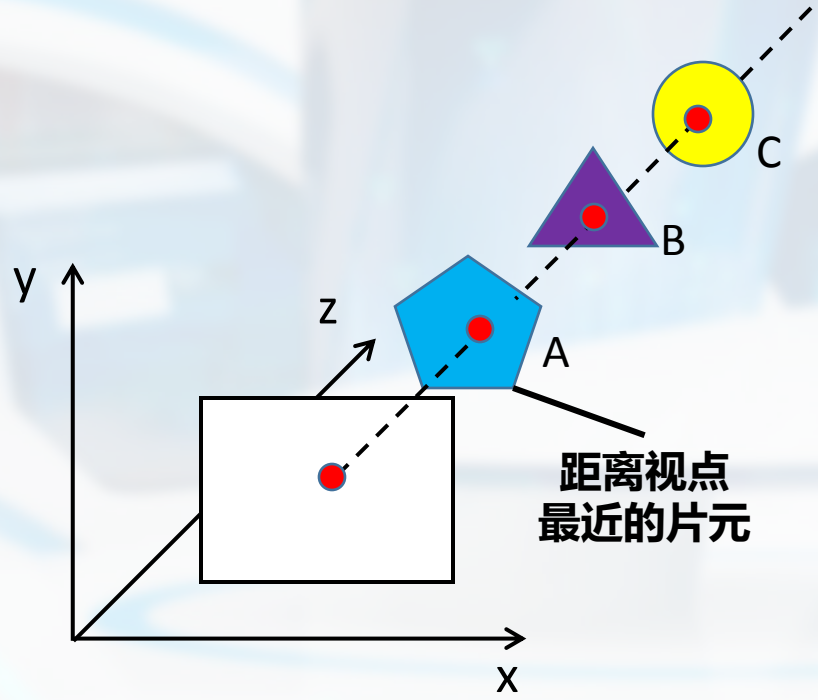
◆深度缓存

深度缓存（Depth Buffer）存储每个片元的深度信息，也就是Z坐标值。



3 深度测试

◆深度缓冲器算法（Z-buffer算法）



算法思想：

对每个像素点找到距离视点最近的片元

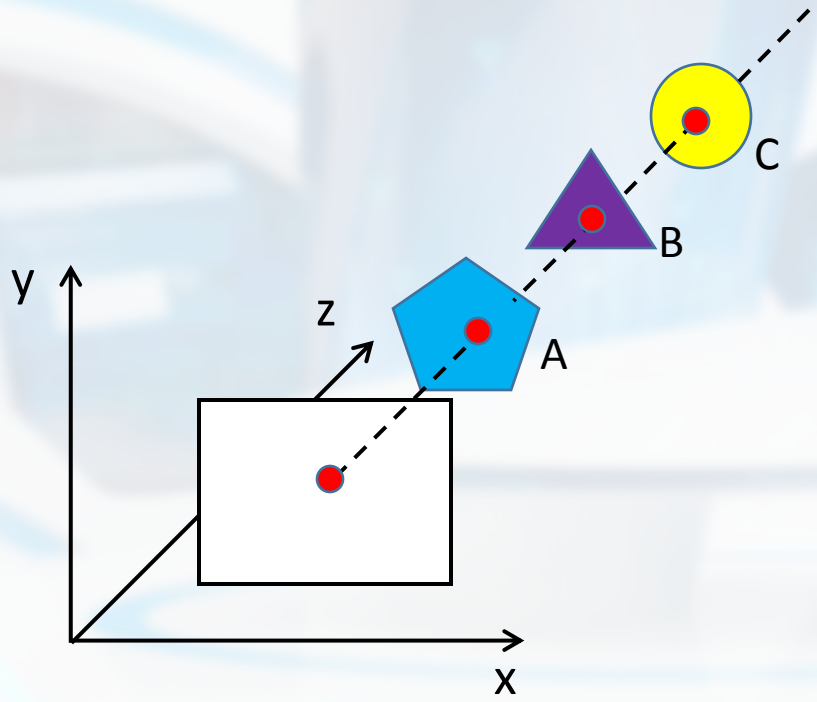
在屏幕空间中：

其实是最靠近坐标屏幕的片元，也就是z值最小的片元

这个片元的颜色值就是这个像素点的颜色值

3 深度测试

◆深度缓冲器算法（Z-buffer算法）



算法步骤：

1、初始化：

将深度缓存与帧缓存中的所有单元(x，y)初始化：

- 深度缓存中各 (x，y)单元置为z的最大值1

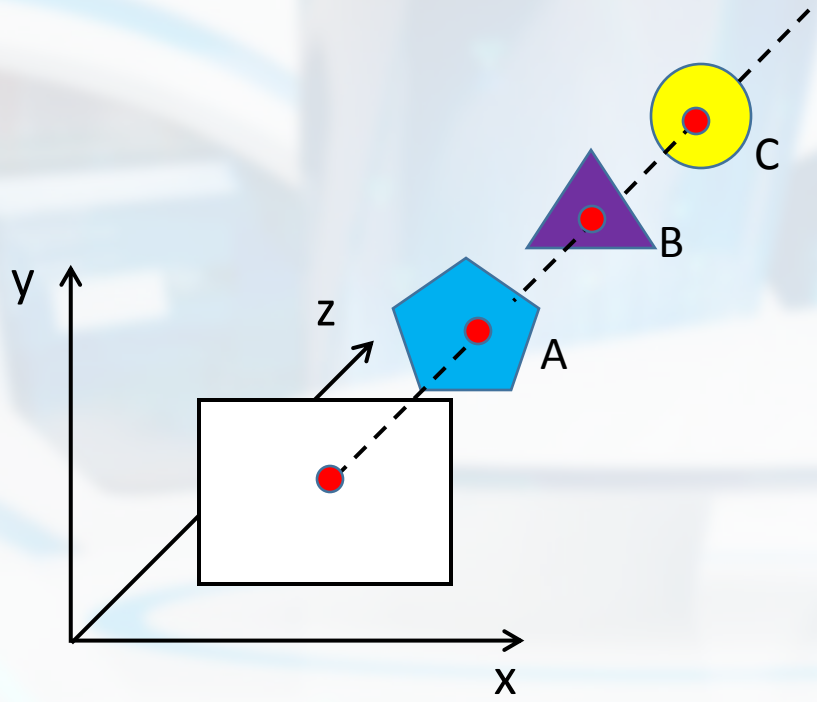
  DepthBuffer（ x，y ）=1

- 帧缓存中各 (x，y)单元颜色值置为背景色

  FrameBuffer（ x，y ）=BackgroudColor

3 深度测试

◆深度缓冲器算法（Z-buffer算法）



算法步骤：

2、处理场景中的每一多边形，每次一个：

计算多边形的上各点(x，y)的深度值z

若z<depthBuff(x，y)

则depthBuff (x，y) =z；

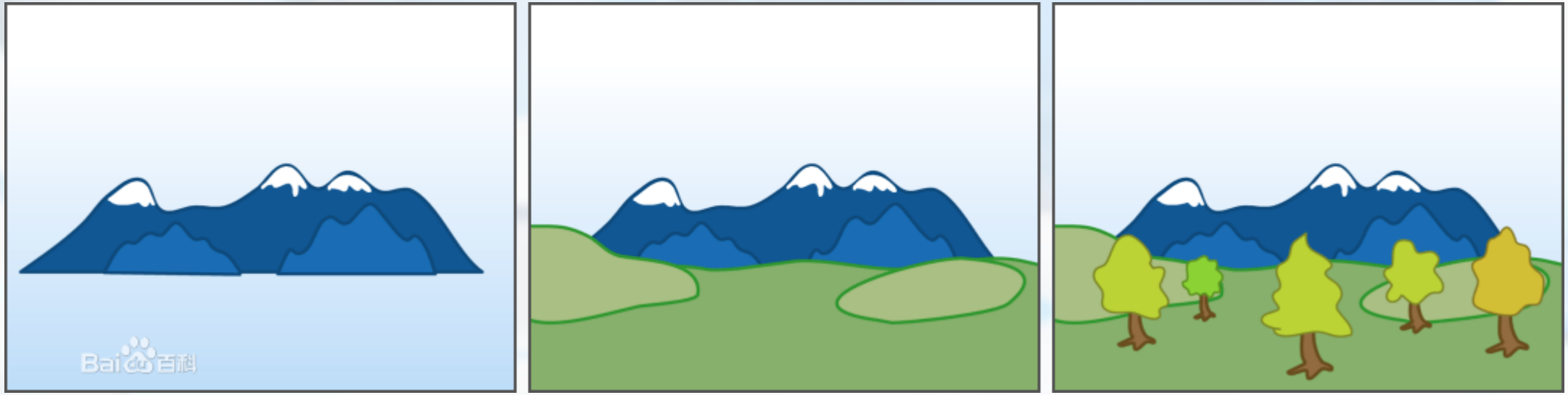
取得该多边形表面的颜色值surfColor (x，y)；

frameBuff (x，y) =surfColor (x，y)

3 深度测试

◆深度排序算法（depth sorting method），又叫画家算法（painter’s algorithm）

思想：画家在创作一幅油画时，总是先画背景，然后画较远处的场景，然后是近一点的物体，最后画最近的景物。



3 深度测试

◆深度排序算法（depth sorting method），又叫画家算法（painter’s algorithm）

数据结构：

（1）多边形队列M：存储所有多边形

（2）优先级队列N：深度排序得到的结果，按优先级存放所有多边形

3 深度测试

◆深度排序算法（depth sorting method），又叫画家算法（painter's algorithm）

算法步骤：

（1）深度排序：将多边形按深度优先级进行排序，结果存入队列N中

距视点近的优先级高，距视点远的优先级低。

（2）扫描转换：从队列N中逐个取出多边形进行绘制

其实就是由优先级低的多边形开始，逐个对多边形进行扫描转换

3 深度测试

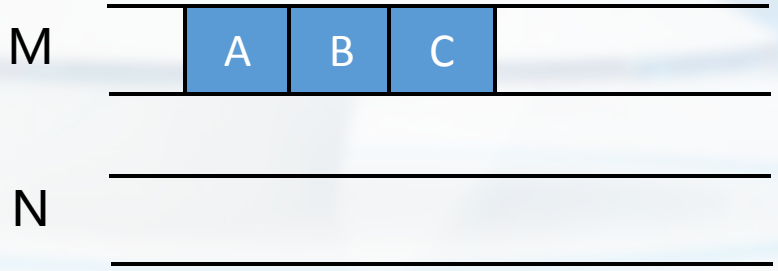
◆深度排序算法（depth sorting method），又叫画家算法（painter's algorithm）

（1）深度排序

Step 1.初始化

将场景中的所有多边形按z_{max}由大到小的顺序存入一个先进先出队列中，记为M；

同时初始化一空的先进先出队列N。



3 深度测试

◆深度排序算法（depth sorting method），又叫画家算法（painter's algorithm）

（1）深度排序

Step 2.只有一个多边形

若M中的多边形个数为1，则将M中的多边形直接加入到N中，同时将A从M中删除。

3 深度测试

◆深度排序算法（depth sorting method），又叫画家算法（painter's algorithm）

（1）深度排序

Step 3. 有多个多边形

从当前M取出多边形B，对A与B进行判别：

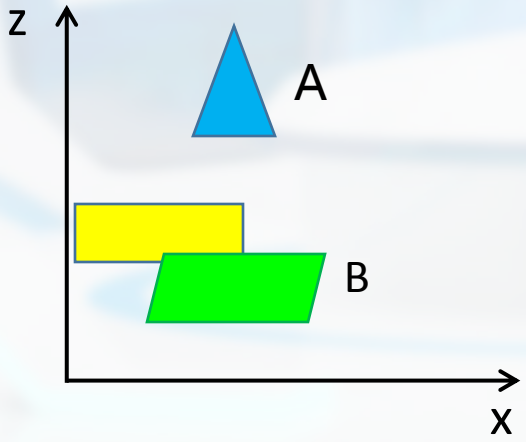
3 深度测试

◆深度排序算法（depth sorting method），又叫画家算法（painter's algorithm）

（1）深度排序

Step 3.有多个多边形

从当前M取出多边形B，对A与B进行判别：



（a）若对M中任意的B均有$z_{max}(B)<z_{min}(A)$，则说明A是M中所有多边形中深度最深的，它与其它多边形在深度方向上无任何重叠，不会遮挡别的多边形。

将A按先进先出原则加入N中。

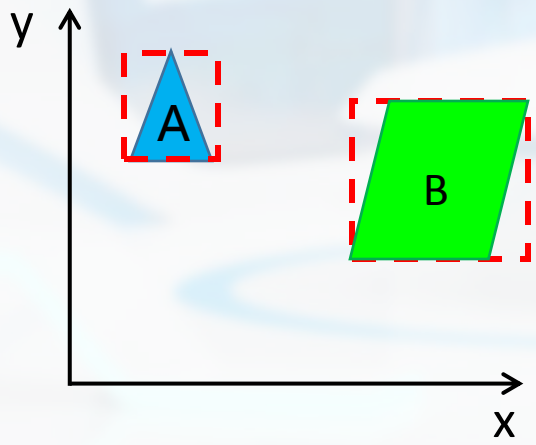
3 深度测试

◆深度排序算法（depth sorting method），又叫画家算法（painter's algorithm）

（1）深度排序

Step 3.有多个多边形

从当前M取出多边形B，对A与B进行判别：



（b）判别多边形A和B在xoy平面上投影的包围盒有无重叠

若无重叠，则A、B在队列中的顺序无关紧要。

将A按先进先出原则加入N中。

3 深度测试

◆深度排序算法（depth sorting method），又叫画家算法（painter's algorithm）

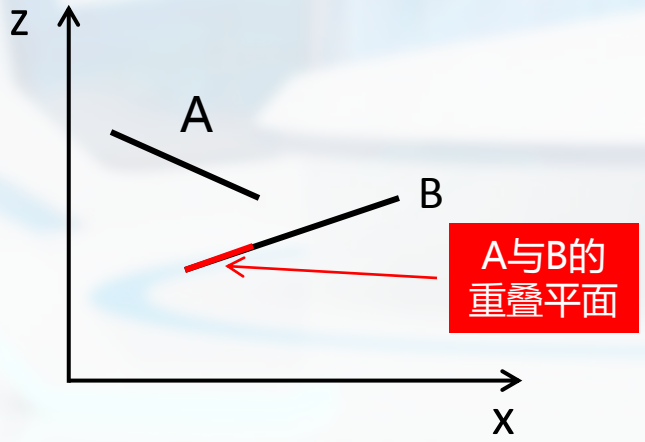
（1）深度排序

Step 3.有多个多边形

从当前M取出多边形B，对A与B进行判别：

（c）判别平面A完全位于B上A与B的重叠平面之后

将A按先进先出原则加入N中。



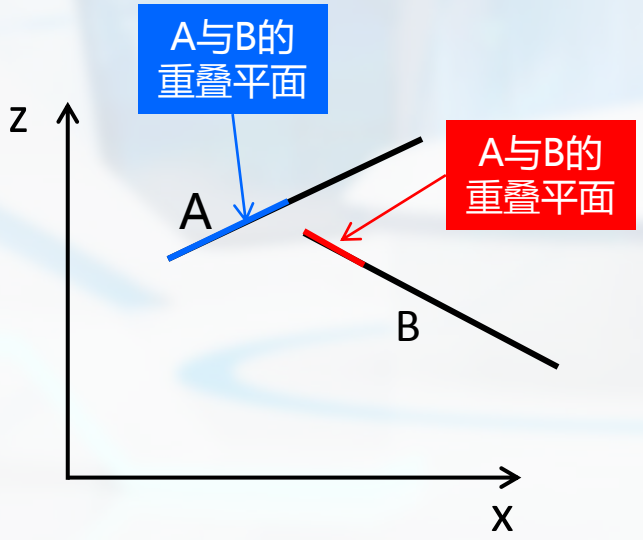
3 深度测试

◆深度排序算法（depth sorting method），又叫画家算法（painter's algorithm）

（1）深度排序

Step 3.有多个多边形

从当前M取出多边形B，对A与B进行判别：



（d）A有部分不在这个重叠面之后：

判别B上平面A与B的重叠平面是否完全位于A之前

若是，将A按先进先出原则加入N中

3 深度测试

◆深度排序算法（depth sorting method），又叫画家算法（painter's algorithm）

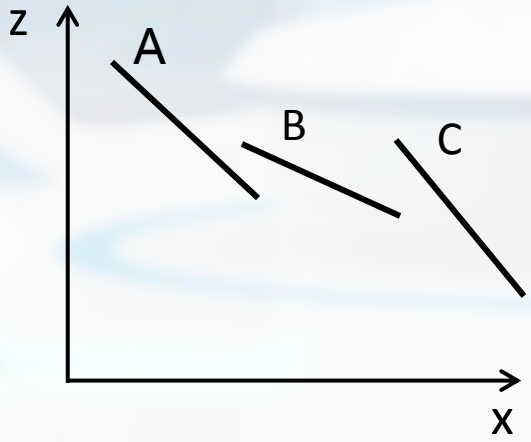
（1）深度排序

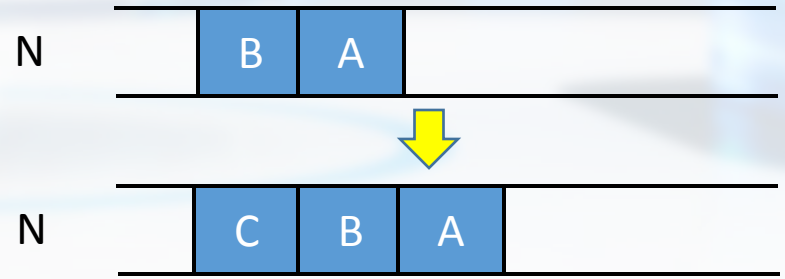
Step 3.当A、B排好序了，继续处理其他的多边形

从当前M取出多边形C：

通过判别发现B遮挡了C的一部分，因此C的优先级最低

最后优先级由低到高的顺序为C、B和A。



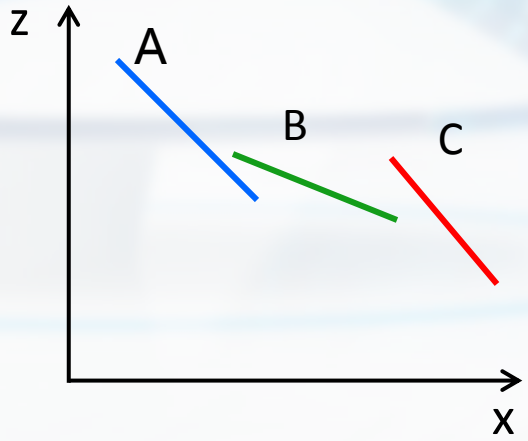


3 深度测试

◆深度排序算法（depth sorting method），又叫画家算法（painter's algorithm）

（2）扫描转换

从N中按照优先级顺序取出多边形进行绘制



4 OpenGL中的消隐

◆面剔除

OpenGL中可以开启多边形剔除

如：glEnable (GL_CULL_FACE) ;

glCullFace (mode) ;

mode可以是GL_FRONT、GL_BACK、GL_FRONT_AND_BACK

◆深度测试

深度测试默认是关闭的，所以如果要启用深度测试的话，我们需要用GL_DEPTH_TEST选项来启用它

如： glEnable(GL_DEPTH_TEST);


谢谢
软件学院　万琳