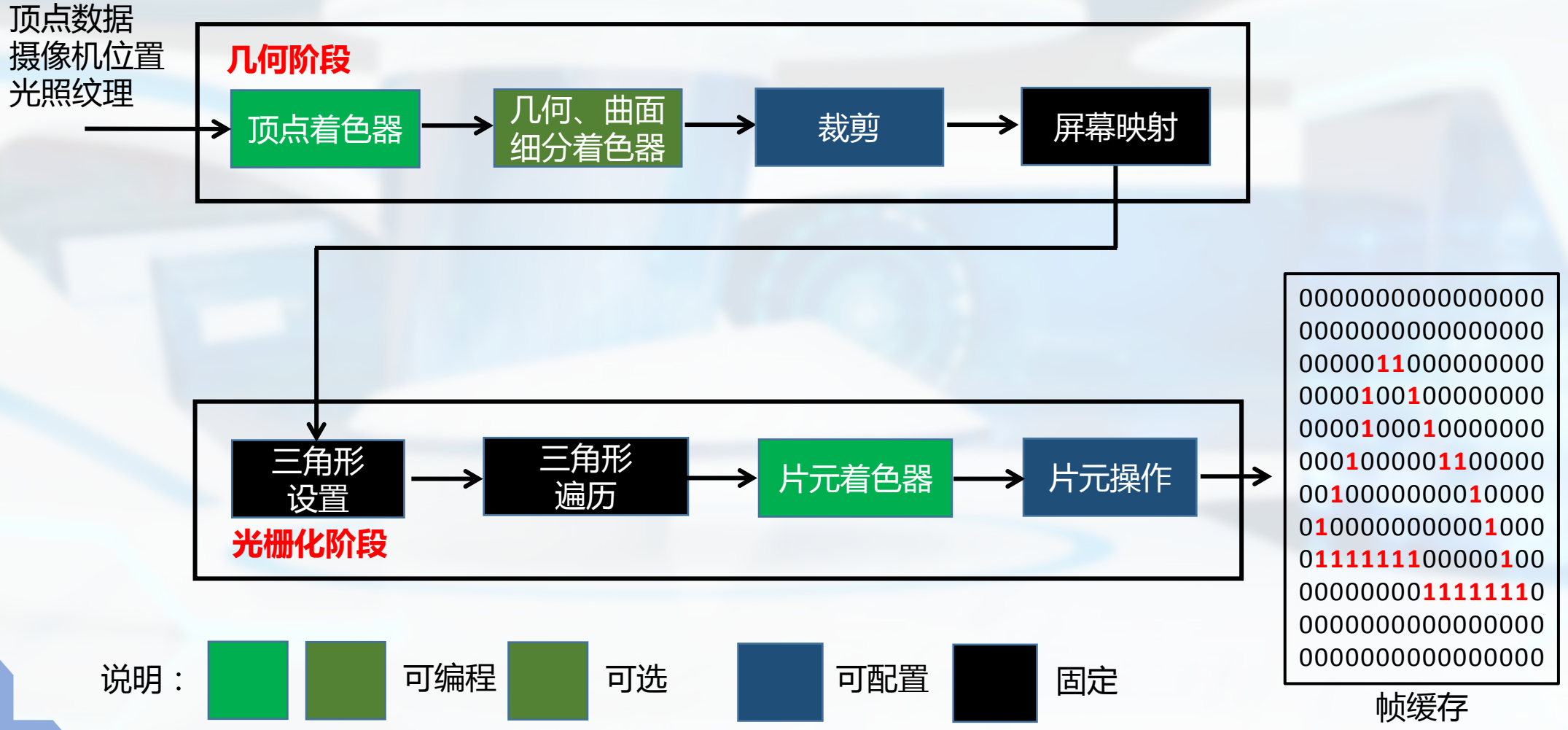
再看片元操作

华中科技大学软件学院　万琳

提纲

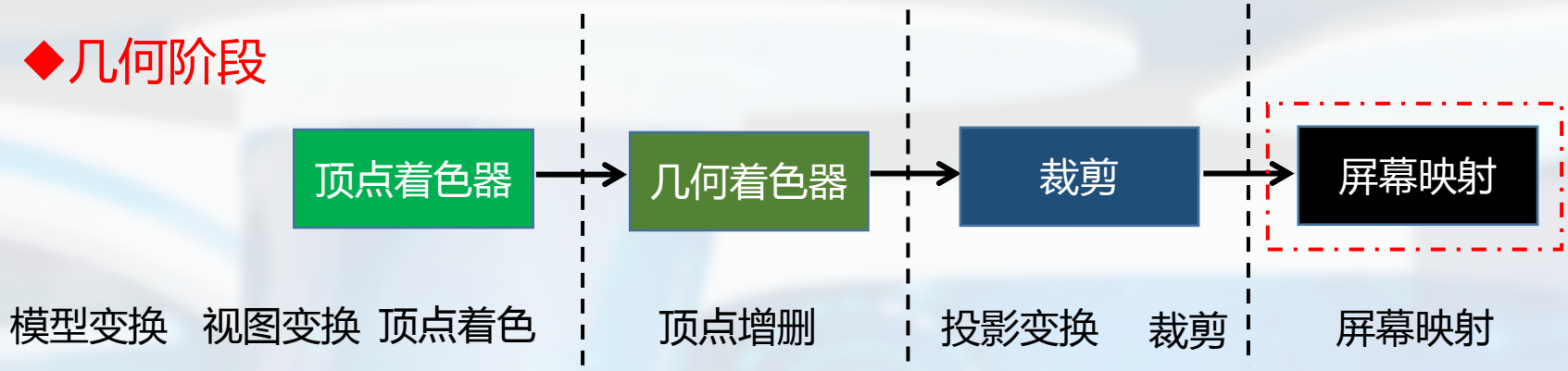
1. 片元操作
2. 几个重要的缓冲区

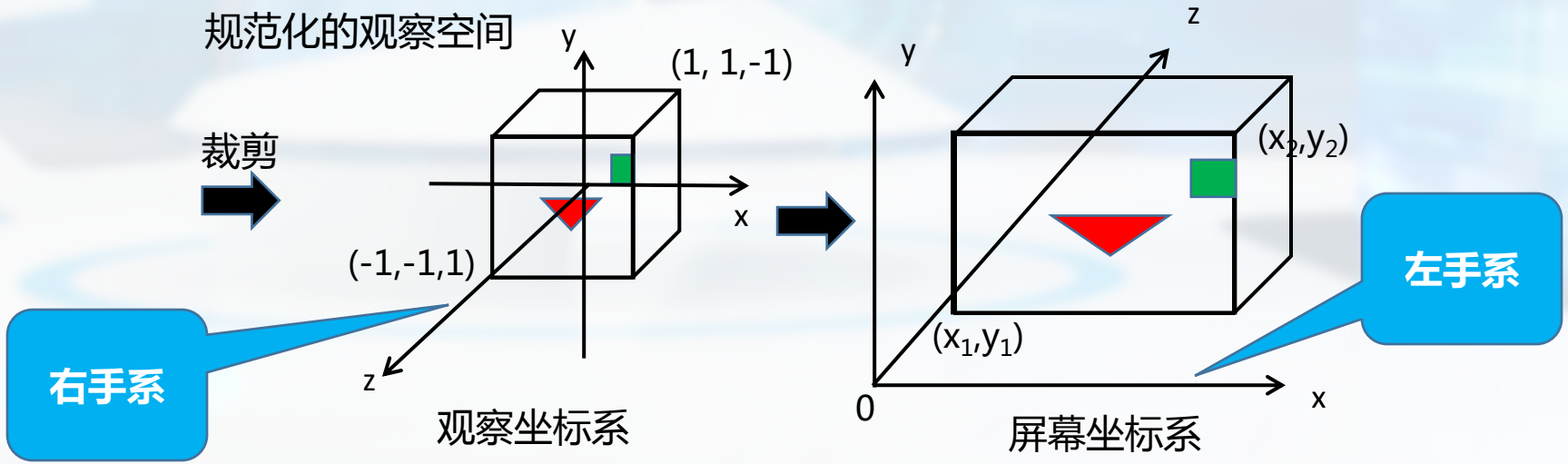
1 片元操作


顶点数据
摄像机位置
光照纹理
几何阶段
顶点着色器
几何、曲面细分着色器
裁剪
屏幕映射
三角形设置
三角形遍历
片元着色器
片元操作
光栅化阶段
0000000000000000
0000000000000000
0000011000000000
0000100100000000
0000100010000000
0001000001100000
0010000000010000
0100000000001000
0111111100000100
0000000011111110
0000000000000000
0000000000000000
帧缓存
说明：
可编程
可选
可配置
固定

1 片元操作

◆几何阶段


顶点着色器
几何着色器
裁剪
屏幕映射
模型变换
视图变换
顶点着色
顶点增删
投影变换
裁剪
屏幕映射

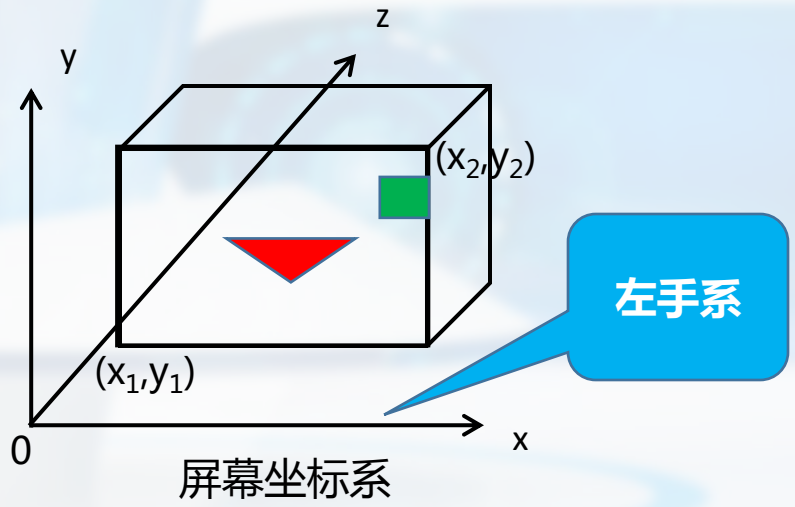

规范化的观察空间
裁剪
y
(1, 1,-1)
x
(-1,-1,1)
z
右手系
观察坐标系
z
y
(x2,y2)
(x1,y1)
0
x
左手系
屏幕坐标系

1 片元操作

◆屏幕坐标系

深度值规范为0~1之间

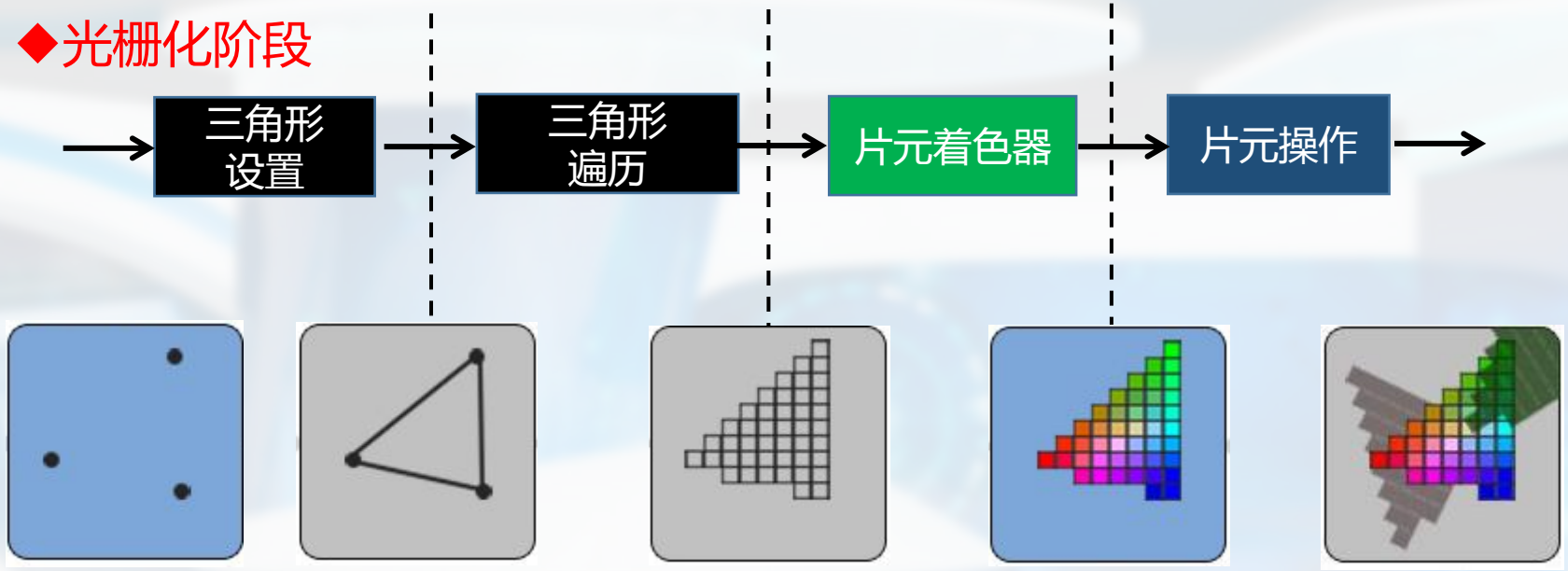
- 近平面：Z=0
- 远平面：Z=1



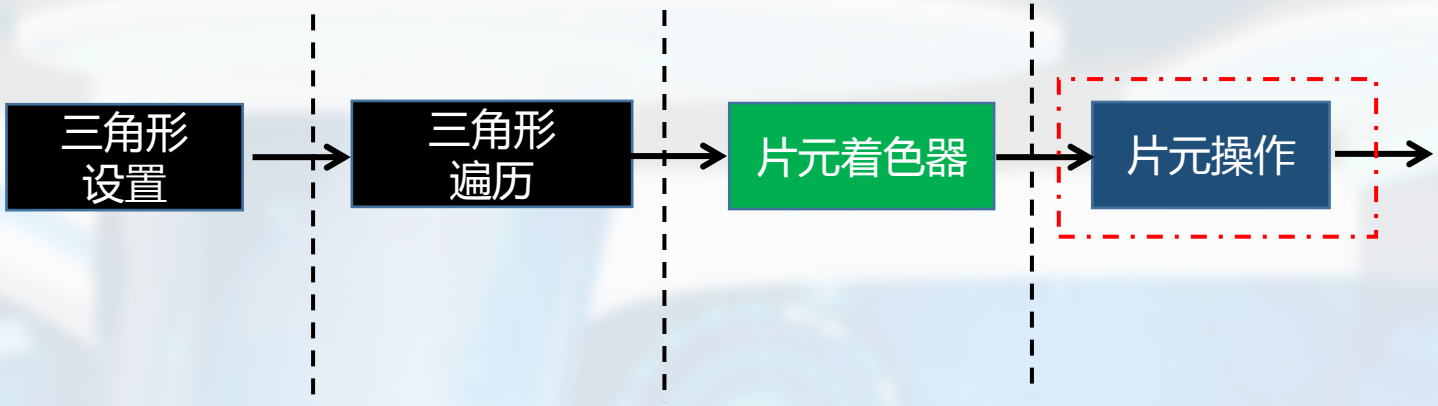
屏幕坐标系

1 片元操作

◆光栅化阶段



1 片元操作



片元操作

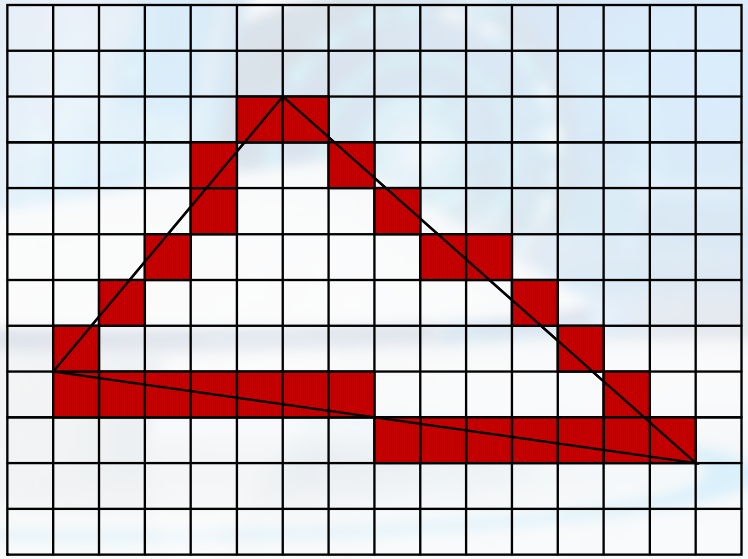
2 几个重要的缓存

- 颜色缓存
- 深度缓存
- 模板缓存
- 累计缓存

2 几个重要的缓冲区

◆颜色缓存

颜色缓存（ Color Buffer/Pixel Buffer ）存储每个像素点的颜色值。



2 几个重要的缓冲区

◆模板缓存

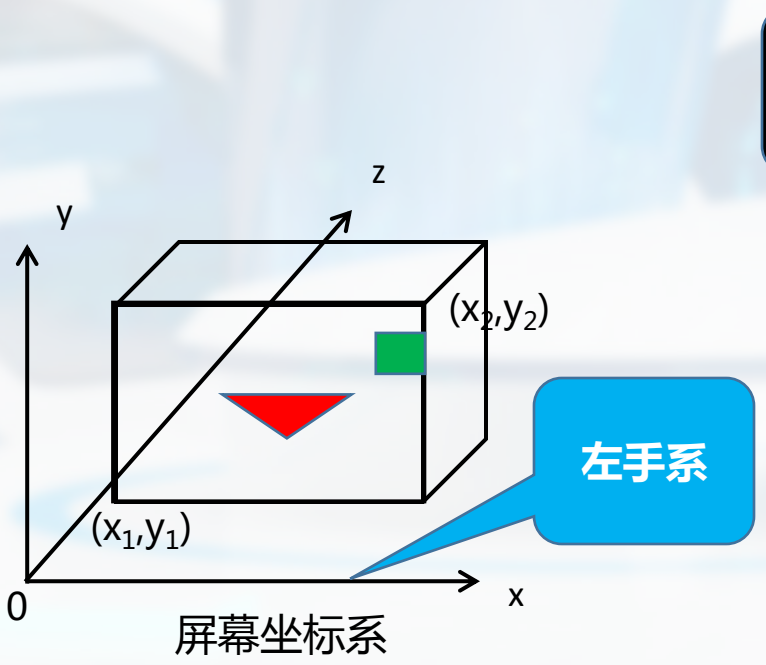
模板缓存（ Stencil Buffer ）存储了一个模板，比如可以设定模板上对应点为1的像素点才会被显示出来。

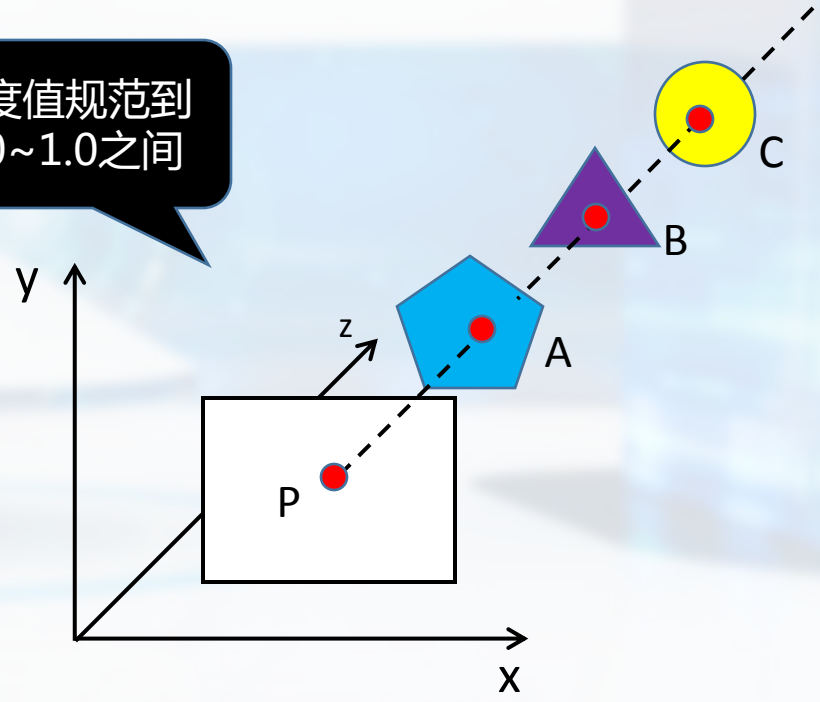
模板缓存

2 几个重要的缓冲区

◆深度缓存

深度缓存（Depth Buffer）存储每个像素点的深度信息，也就是Z坐标值。





2 几个重要的缓冲区

◆累计缓存

累积缓存（Accumulation Buffer）：与颜色缓冲类似，同样储存像素点的颜色值。

用途：合成多幅图像，实现在场景中“多重曝光（multiple exposures）” 。

方法：通过将图像渲染多次，对场景位置（或所选的物体）进行微小的、渐增的改变，然后累积结果。

效果：提高图像的真实性，产生反走样、运动模糊等效果。

2 几个重要的缓冲区

◆累计缓存

运动模糊：多幅有微小位移的图像的合成。

3 再看片元操作

片元 → **模板测试** → 深度测试 → 颜色混合 → 颜色缓冲区

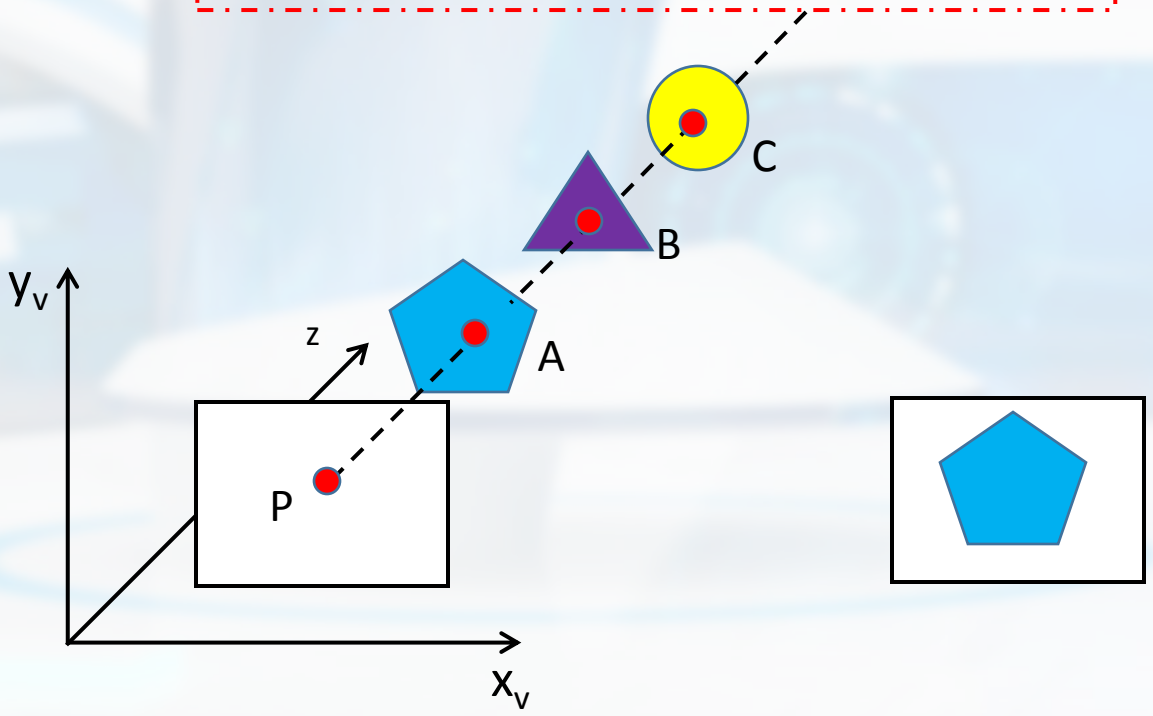
模板缓存

3 再看片元操作

片元 → **模板测试** → 深度测试 → 颜色混合 → 颜色缓冲区

3 再看片元操作

片元 → 模板测试 → **深度测试** → 颜色混合 → 颜色缓冲区

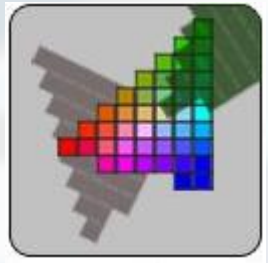


3 再看片元操作

什么是颜色混合？

3 再看片元操作

颜色混合（Color Blending）



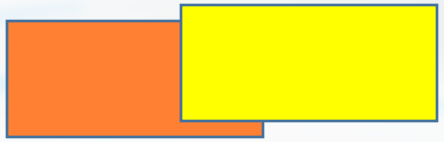
当A（实际上是α系数，Alpha Coeefficient）不为1.0f，
即颜色有一定透明度时，可以进行颜色混合

RGBA源颜色值（R_s,G_s,B_s,A_s），RGBA目标颜色值（R_d,G_d,B_d,A_d）

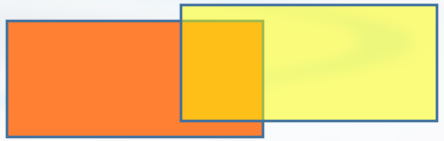
源调和因子（S_r,S_g,S_b,S_a），　目标调和因子（D_r,D_g,D_b,D_a）

新的调和颜色计算：

$$S_rR_s +D_rR_d,S_gG_s +D_gG_d,S_bB_s +D_bB_d,S_aA_s +D_aA_d$$



不透明时不进行颜色混合



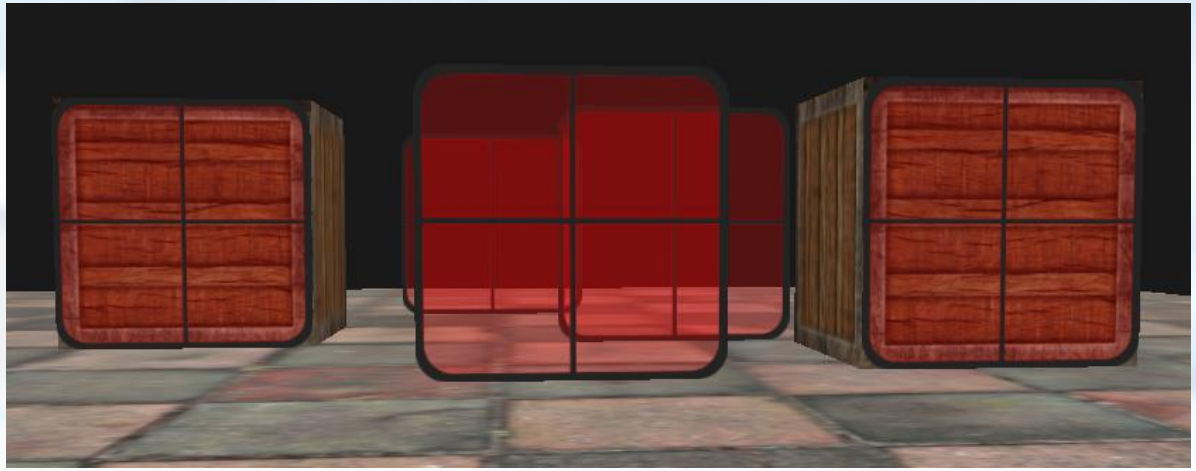
一个透明时的颜色混合

两个都透明时的颜色混合

3 再看片元操作

片元 → 模板测试 → 深度测试 → **颜色混合** → 颜色缓冲区

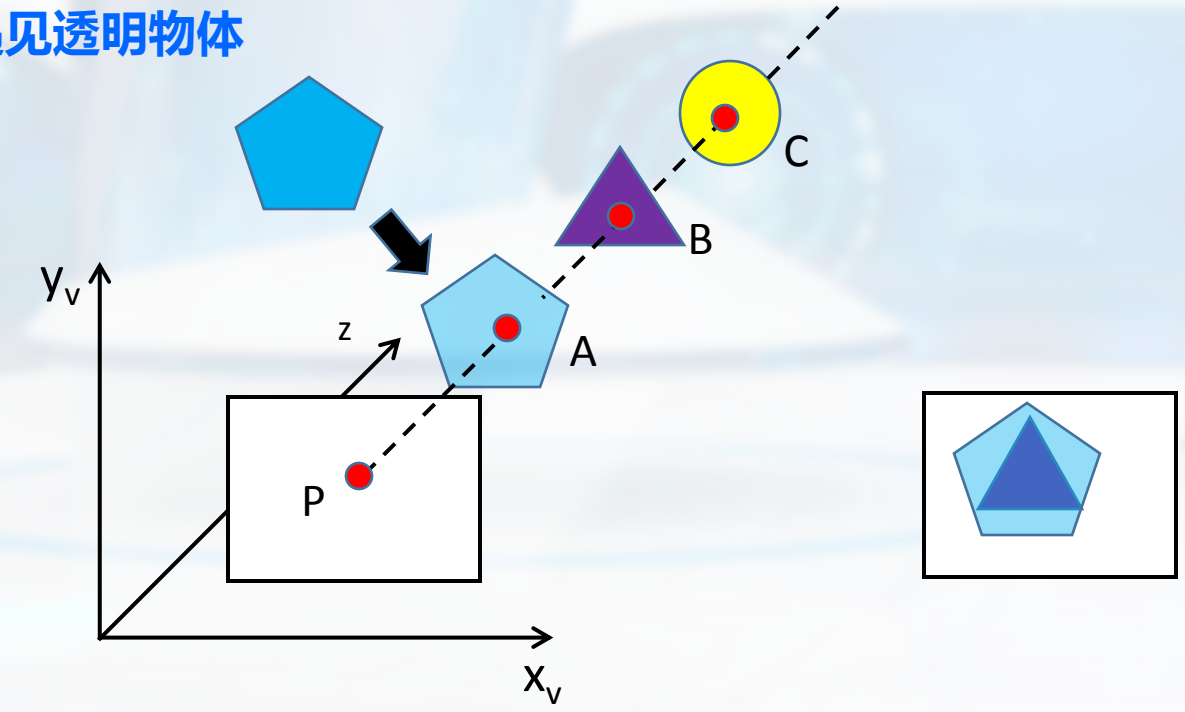
遇见透明物体



3 再看片元操作

片元 → 模板测试 → 深度测试 → **颜色混合** → 颜色缓冲区

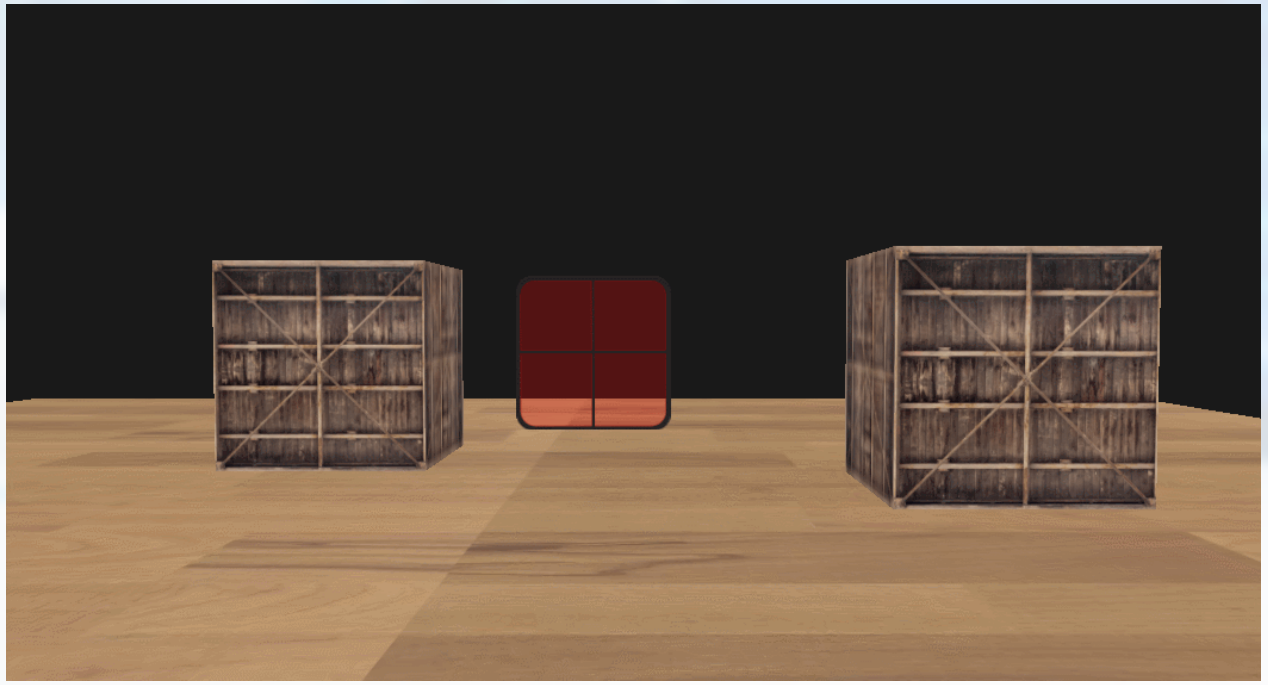
遇见透明物体



3 再看片元操作

片元 → 模板测试 → 深度测试 → **颜色混合** → 颜色缓冲区

遇见透明物体



3 再看片元操作

片元 → 模板测试 → 深度测试 → **颜色混合** → 颜色缓冲区

处理某些特效：如运动模糊

3 再看片元操作

片元 → 模板测试 → 深度测试 → **颜色混合** → 颜色缓冲区

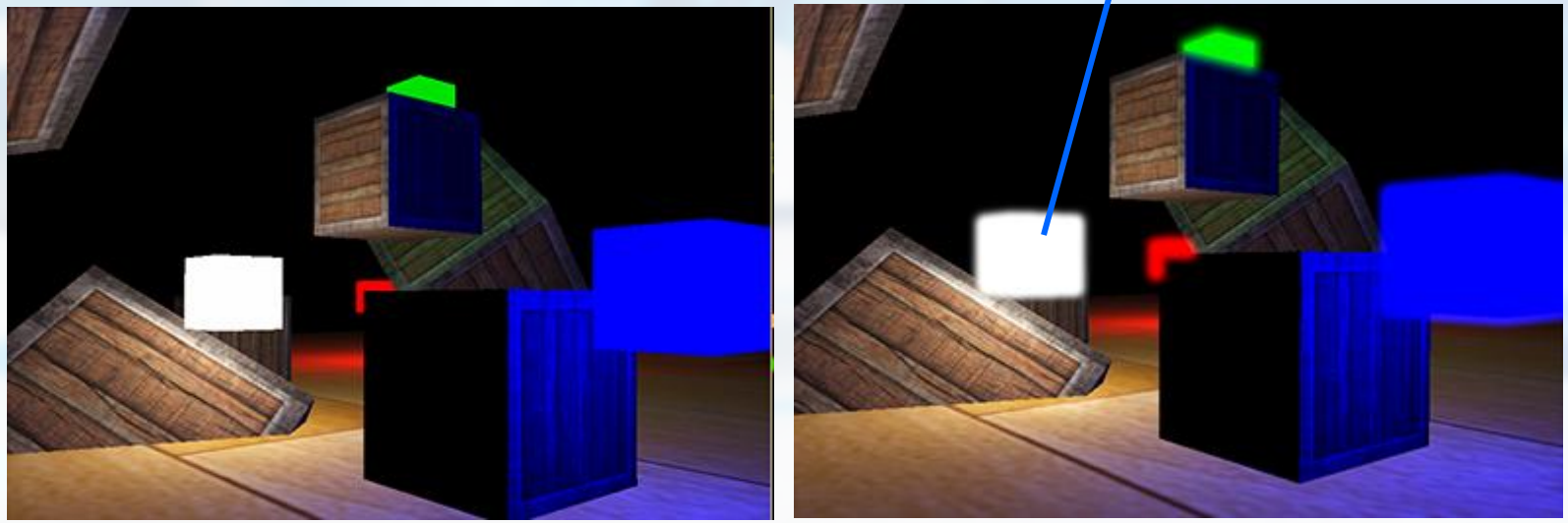
处理某些特效：如运动模糊

合并后的效果

3 再看片元操作

片元 → 模板测试 → 深度测试 → **颜色混合** → 颜色缓冲区

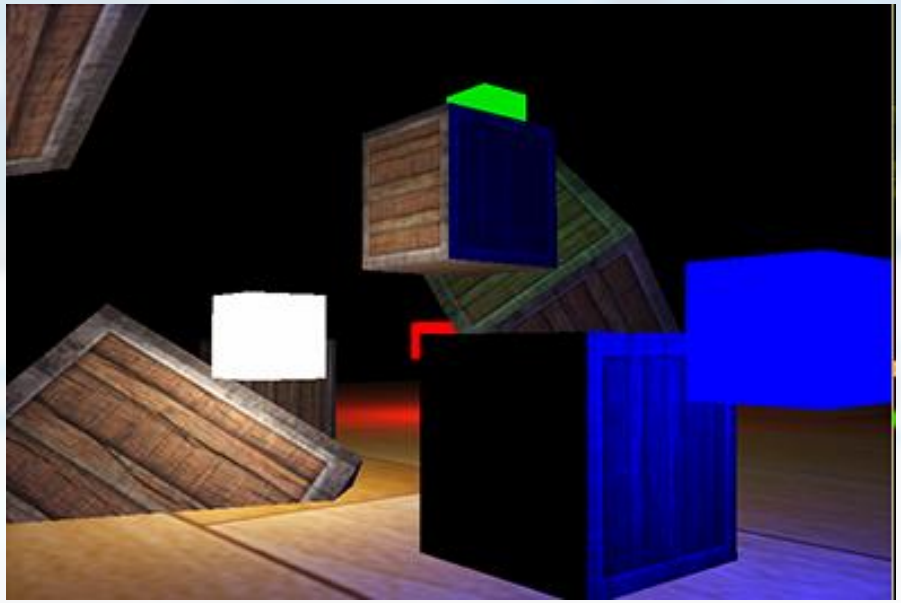
处理某些特效：如泛光效果Bloom

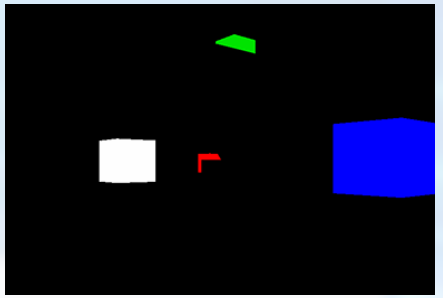


3 再看片元操作

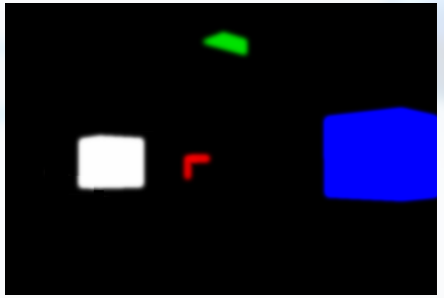
片元 → 模板测试 → 深度测试 → **颜色混合** → 颜色缓冲区

处理某些特效：如泛光效果Bloom





提取场景中光亮的部分

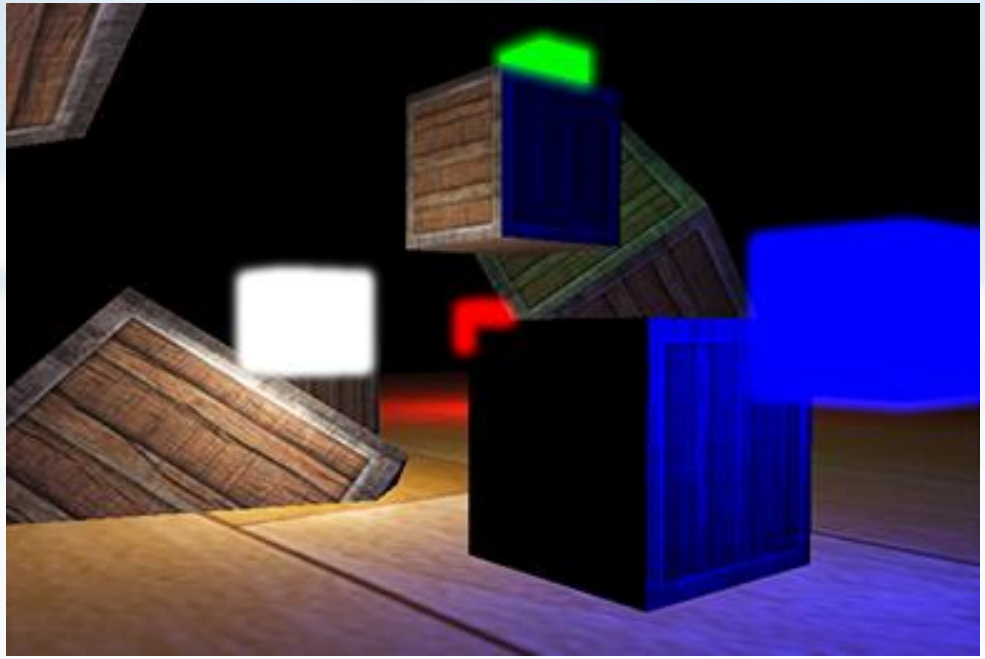


进行模糊，如高斯模糊

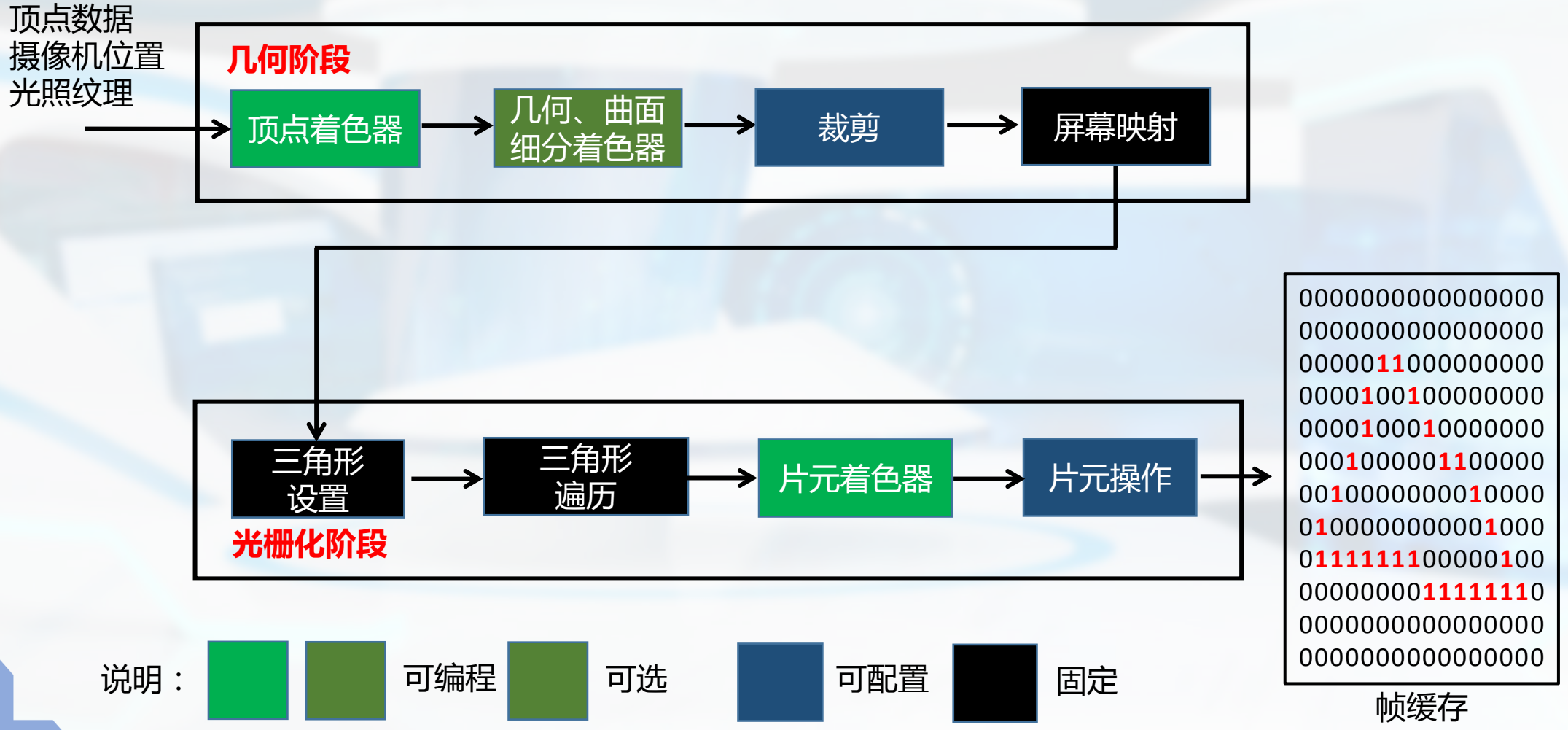
3 再看片元操作

片元 → 模板测试 → 深度测试 → **颜色混合** → 颜色缓冲区

处理某些特效：如泛光效果Bloom

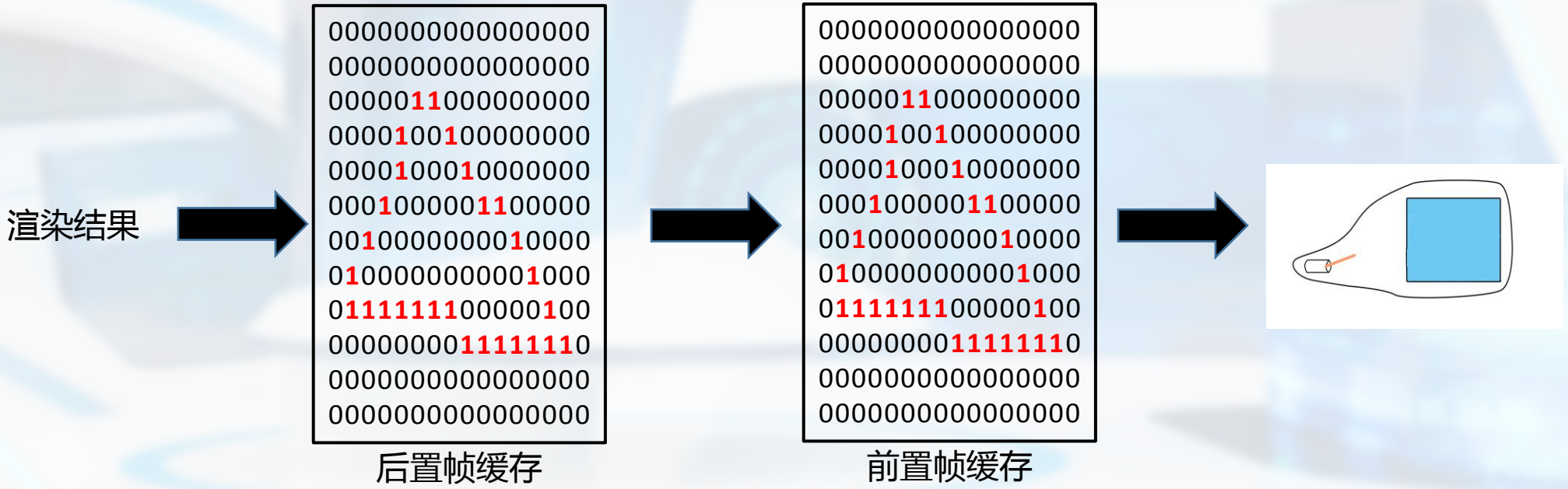


3 再看片元操作


顶点数据
摄像机位置
光照纹理
几何阶段
顶点着色器
几何、曲面细分着色器
裁剪
屏幕映射
三角形设置
三角形遍历
片元着色器
片元操作
光栅化阶段
0000000000000000
0000000000000000
0000011000000000
0000100100000000
0000100010000000
0001000001100000
0010000000010000
0100000000001000
0111111100000100
0000000011111110
0000000000000000
0000000000000000
帧缓存
说明：
可编程
可选
可配置
固定

3 再看片元操作

◆GPU的双重缓冲（Double Buffering）



3 再看片元操作

10.2谁遮住了我？


谢谢
软件学院　万琳