计算机图形学小白入门

————从0开始实现OpenGL

双线性插值采样

授课：赵新政
资深三维工程师

专注3D图形学技术
教育品牌

双线性插值采样

纹理失真

- 在使用纹理坐标绘制三角形的时候，当图片宽/高（**分辨率**）不够的时候，会出现明显的颗粒感

双线性插值采样

纹理原因

- 采样不足举例：

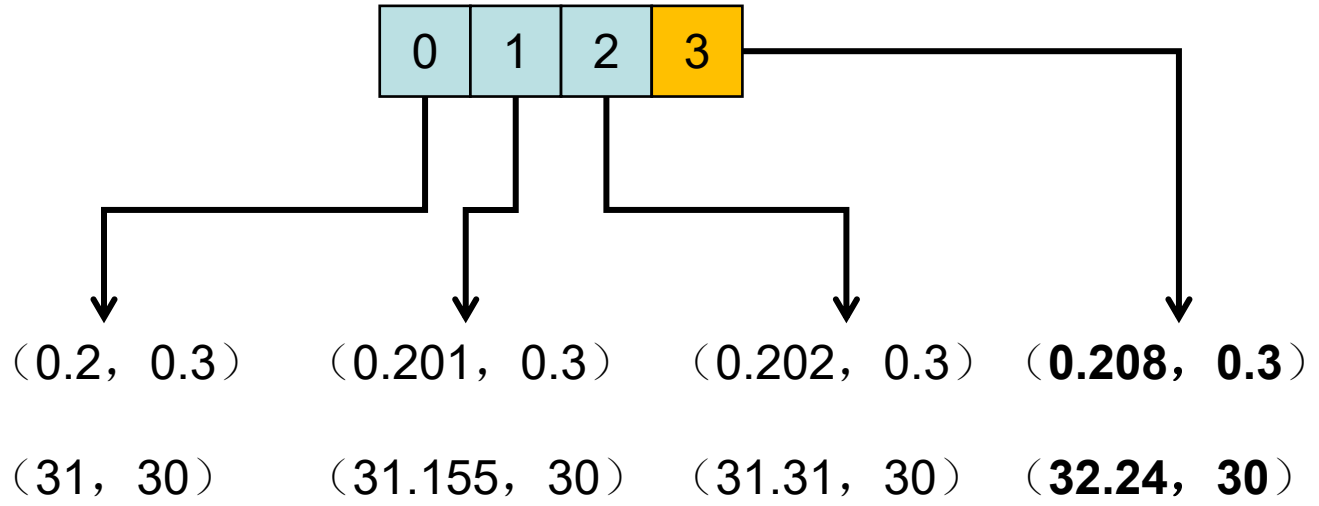
  如果图片宽高为：**200*250**，而需要绘制的矩形为**400*500**，那么多达**20万个屏幕像素点**需要用**5万个图片像素**表达

- 此种情况下，称为过采样（Over-Sampling），即**需要的数据比源头数据要多**
- 数据不足的情况下，出现的图像也就更加的颗粒感/粗糙
- 多个屏幕像素点都只能用**临近的同一个图片像素**进行上色，很类似像素风的游戏

双线性插值采样

过采样演示

- 假设被采样的图像为155 * 100的大小



0/1/2屏幕像素点，经过四舍五入后的结果都是（**31，30**），那么就会采样到同样的像素点

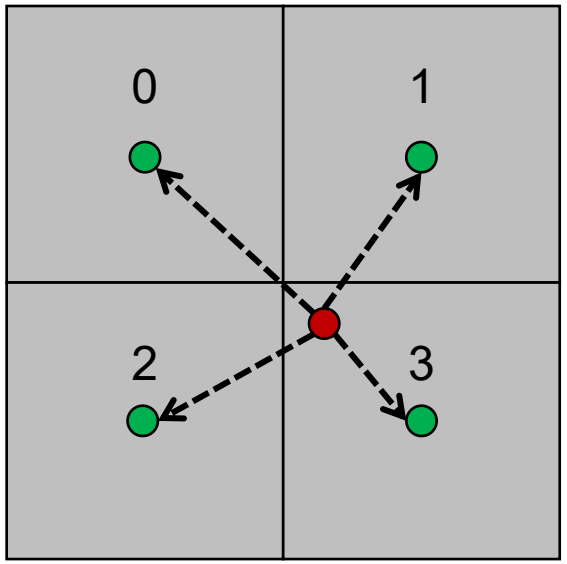
3号四舍五入后是（**32，30**）的像素点

他们就会产生“**突变**”，不平滑

诉求：我们应该寻求一种方法，在图片像素不足的情况下，“**产生**”新的图片像素给到每个屏幕位置

双线性插值采样

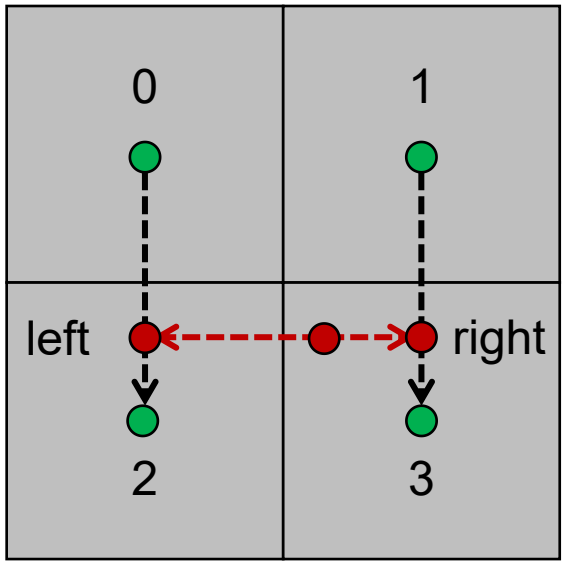
- **双线性采样思想**
- uv坐标计算出来的采样位置都是带有**小数点的数据**，可以选择周边四个像素进行差值



如何决定四个点的权重及插值方法，就是双线性插值采样的基础工作

双线性插值采样

- **双线性采样思想**
- uv坐标计算出来的采样位置都是带有**小数点的数据**，可以选择周边四个像素进行差值

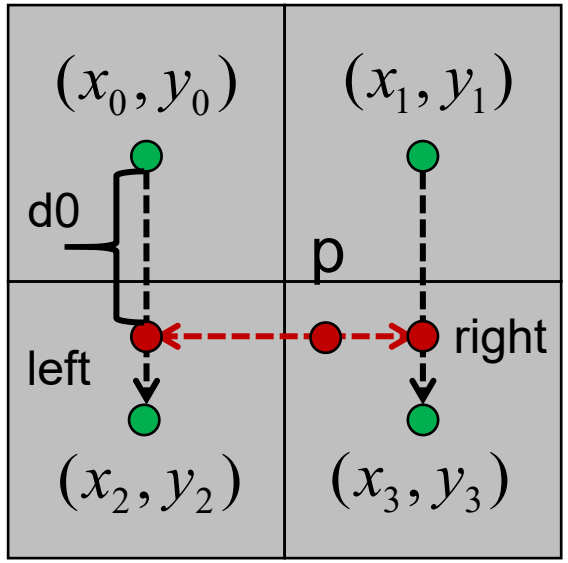


- **一次线性插值**
- 使用0号点与2号点，插值颜色给到left
- 使用1号点与3号点，插值颜色给到right

- **二次线性插值**
- 使用left与right的颜色进行插值给到目标点

双线性插值采样

- **双线性采样算法**
- 已知p点的坐标为 (x_p, y_p)，其余像素点坐标如下所示



$$d_0 = y_0 - y_p$$

$$d = y_0 - y_2$$

$$yScale = \frac{d_0}{d}$$

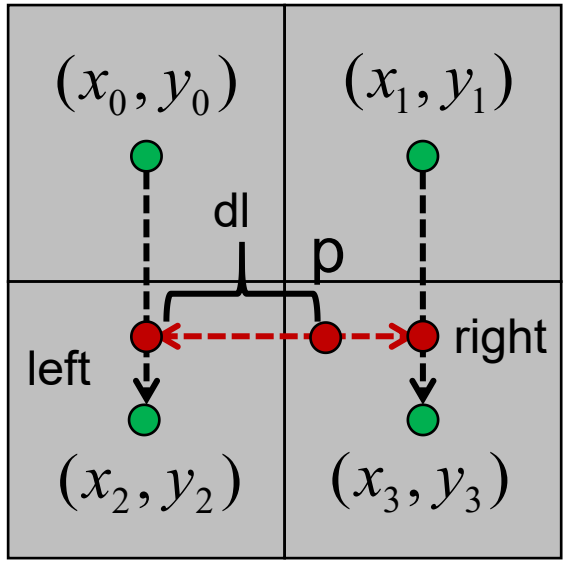
$$leftColor = c_2.yScale + c_0.(1 - yScale)$$

同理，对于right点的颜色，可以表达如下：

$$rightColor = c_3.yScale + c_1.(1 - yScale)$$

双线性插值采样

- **双线性采样算法**
- 已知left与right颜色，使用插值算法求取最终p点颜色



$$dl = x_p - x_2$$

$$d = x_3 - x_2$$

$$xScale = \frac{d_0}{d}$$

$$color_p = c_{right}.xScale + c_{left}.(1 - xScale)$$

在使用了两次线性插值之后，我们就可以使用p点周围四个像素，“产生”了p点的新颜色