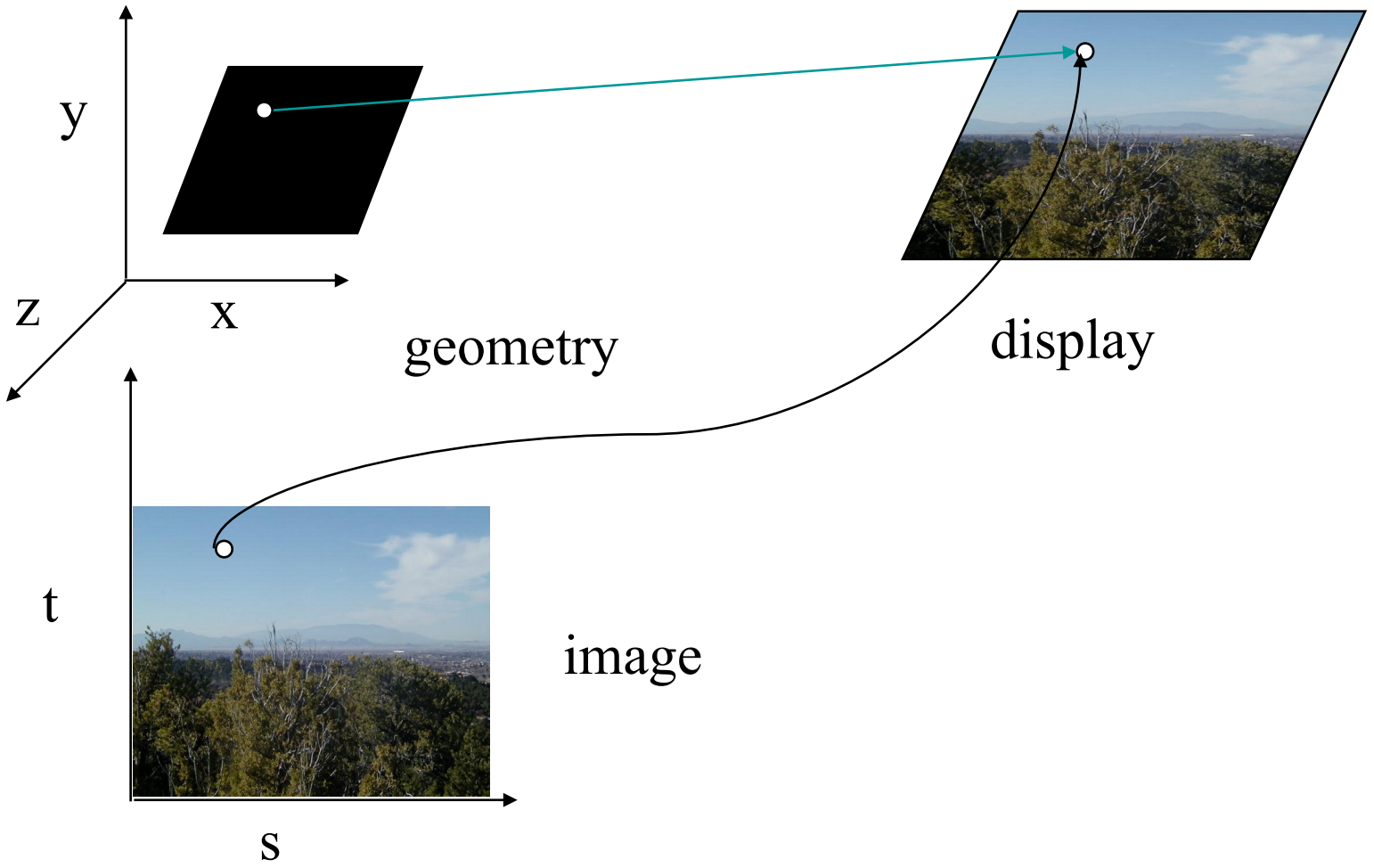
纹理

纹理映射

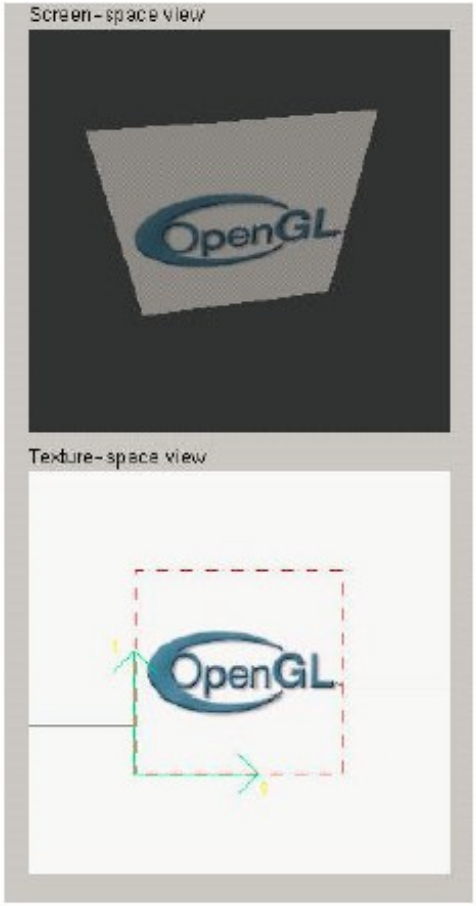


基本步骤

- 指定纹理
  - 读入或生成纹理图像
  - 关联纹理数据
  - 激活纹理映射功能
- 指定纹理参数
  - wrapping, filtering
- 为每个顶点指定纹理坐标

示例

- 纹理为 256 × 256 的图像，它被映射到一个矩形上
- 经透视投影后的结果



指定纹理数据（图像）

- GLubyte my_texels[512][512];
- 定义纹理图像所用的数据
  - –图像
  - –由应用程序代码创建
- 激活纹理映射
- glEnable(GL_TEXTURE_2D);

定义纹理所用的图像

- glTexImage2D(target,level,components,w,h,border,format,type,texels);
  - –target: 纹理的类型： GL_TEXTURE_2D
  - –level:mipmapping
  - –components: 每个纹理元素的分量数
  - –w,h:texels 中以像素为单位的宽度与高度
  - –border: 光滑处理； format 与 type: 描述纹理元素
  - –texels: 指向纹理元素数组的指针
- glTexImage2D(GL_TEXTURE_2D,0,3,512,512,0,GL_RGB,GL_UNSIGNED_BYTE,my_texels)

转化纹理图像

- OpenGL 需要纹理的尺寸为 2 的幂次
- 　如果图像的尺寸不是 2 的幂次，用下述函
- 数进行转化
- gluScaleImage(format,w_in,h_in,

type_in,*data_in,w_out,h_out,

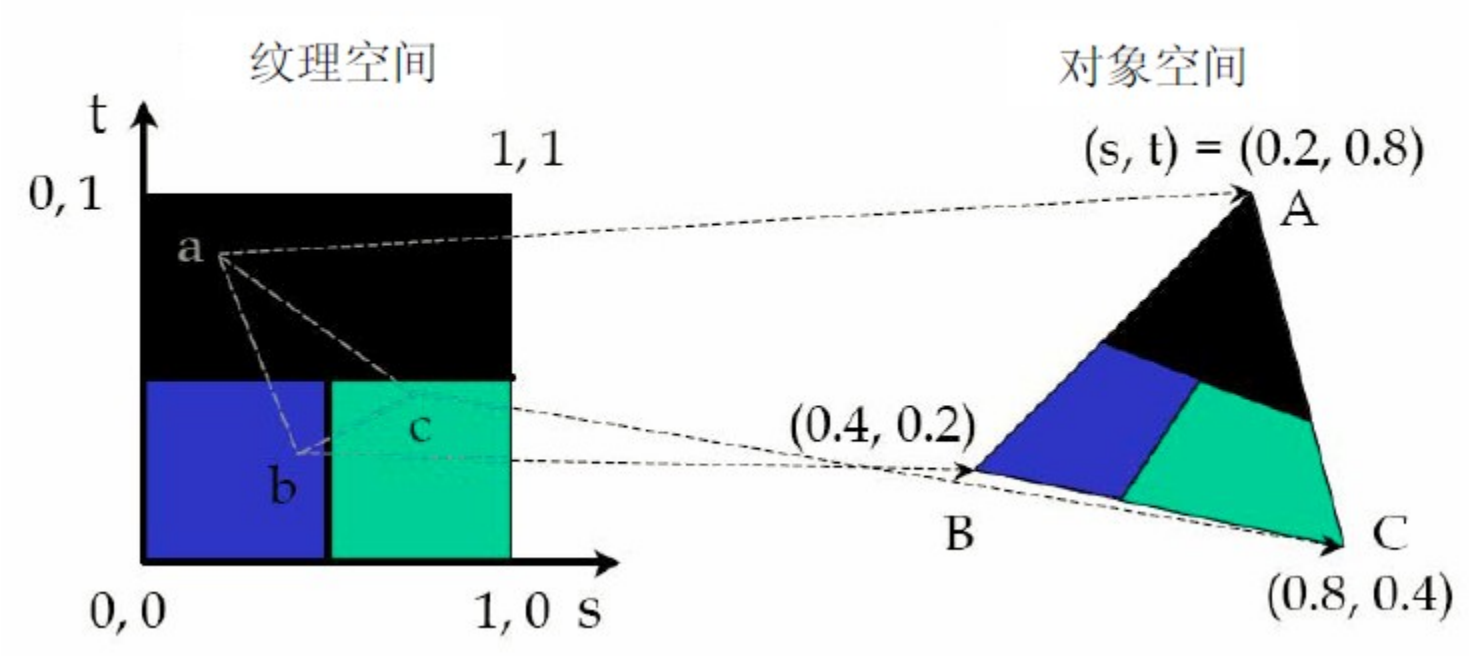
type_out,*data_out);

data_in 源图像

data_out 目标图像

映射纹理

- glTexCoord*() 指定每个顶点对应的纹理坐标



实例代码

- 例

```
glBegin(GL_POLYGON);

glColor3f(r0,g0,b0);
glNormal3f(u0,v0,w0);
glTexCoord2f(s0,t0);
glVertex3f(x0,y0,z0);

glColor3f(r1,g1,b1);
glNormal3f(u1,v1,w1);
glTexCoord2f(s1,t1);
glVertex3f(x1,y1,z1);
.
.
glEnd();
```

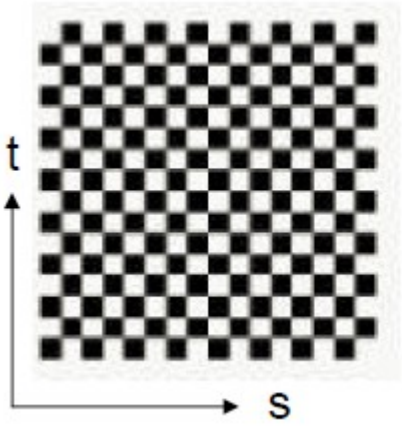
- 插值
  - –OpenGL 应用双线性插值从给定的纹理坐标中求出适当的纹理元素

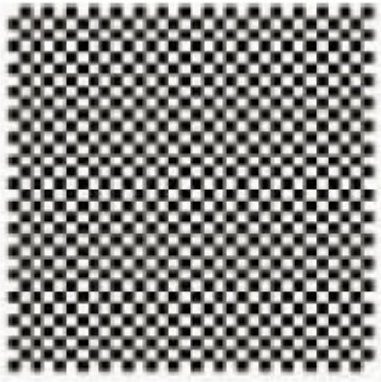
纹理参数

- 确定纹理的使用方式
- Wrapping 参数确定当 s, t 的值超出 [0,1] 区间后的处理方法
- 应用 filter 模式就会不采用点取样方法，而是采用区域平均方法
- Mimmapping 技术使得能以不同的分辨率应用纹理
- 环境参数确定纹理映射与明处理的交互作用

Wrapping 模式

- 截断：若 s,t>1 就取 1 ，若 s,t<0 就取 0
- 重复：应用 s,t 模 1 的值
  - –glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_WRAP_S,GL_CLAMP)
  - –glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_WRAP_T,GL_REPEAT)

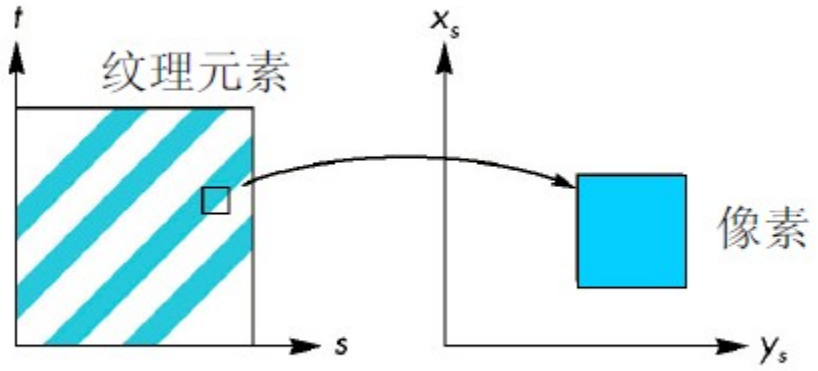



GL_REPEAT

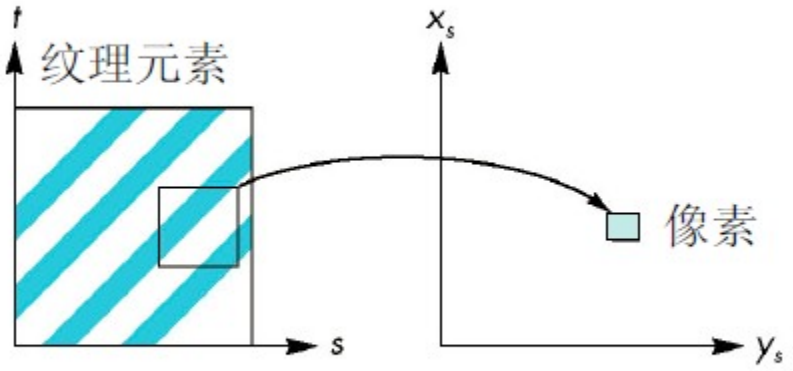

GL_CLAMP

滤波模式：

- 可以是多个纹理元素覆盖一个像素(缩小)，也可以是多个像素覆盖一个纹理元素(放大)



(a)



(b)

指定滤波模式

- •glTexParameteri(target, type, mode)
- •glTexParameteri(GL_TEXTURE_2D,
- GL_TEXTURE_MAG_FILTER, GL_NEARE ST);
- •glTexParameteri(GL_TEXTURE_2D,
- GL_TEXTURE_MIN_FILTER, GL_LINEAR);
- 注意在线性滤波中为滤波边界需要纹理元素具有额外的边界 (border=1)

Mipmap 纹理

- Mipmap 对纹理位图进行预先滤波，降低分辨率；可以减小对于非常小的要加纹理的对象的插值误差
- 在纹理定义时声明 mipmap 的层次
- glTexImage2D(GL_TEXTURE_*D, level, …);
- GLU ：可以从给定图像建立起所有的 mipmap 纹理
- gluBuild*DMipmaps(…)

控制纹理的应用方式

- glTexEnv{fi}[v](GL_TEXTURE_ENV, mode, param);
- GL_TEXTURE_ENV_MODE 设置模式
  - GL_MODULATE: 与计算的明暗效果进行调制
  - GL_BLEND: 与环境颜色融合在一起
  - GL_REPLACE: 只应用纹理的颜色
  - GL_DECAL: 与 GL_REPLACE 相同
- 应用 GL_TEXTURE_ENV_COLOR 设置融合颜色

纹理坐标的自动生成

- 自动生成纹理坐标

glTexGen{ifd}[v](GLenum c, GLenum pn, pv)

• c: 纹理坐标，可取 GL_S, GL_T, GL_R, GL_Q

• pn: 纹理坐标的生成函数，可取 GL_TEXTURE_GEN_MODE, GL_OBJECT_PLANE, GL_EYE_PLANE

• pv: 参数值，当 pn=GL_TEXTURE_GEN_MODE 时，可取 GL_OBJECT_LINEAR, GL_EYE_LINEAR, GL_SPHERE_MAP ；否则应为数组，由生成函数的系数组成

绑定纹理对象

- 过程

1. 生成纹理 ID ： glGenTextures(1, &texture[0]);
2. 创建纹理对象
glBindTexture(GL_TEXTURE_*D, texture[0]);

3. 指定纹理内容： glTexImage*D();
4. 指定纹理属性： glTexParameteri(…, ..., …)

5. 激活纹理功能
glBindTexture(GL_TEXTURE_*D, texture[0]);
6. 提供纹理坐标： glTexCoord*()