

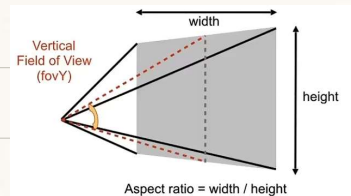
第五课 光栅化(三角形)

一、压缩变换(续)

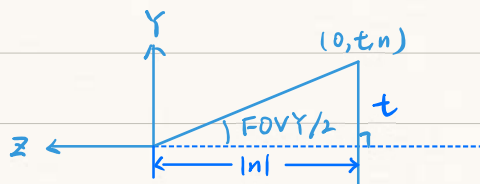
1. 对于近平面的定义 l (左), r (右), b (下), t (上)

(1) field-of-view (可视角度)(FOV)

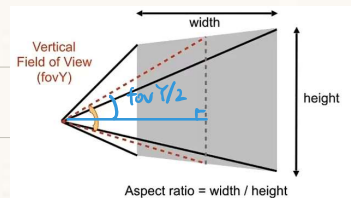
(2) aspect ration: 长宽比



2. FOV 和 l, r, b, t 的转换



$$\tan \frac{fov_Y}{2} = \frac{t}{|n|}$$
$$aspect = \frac{r}{t}$$



二、从立方体到屏幕

1. 什么是屏幕.

(1) 一个二维数组: 像素

(2) 数组的大小: 分辨率

(3) 一个典型的光栅设备

2. 光栅与光栅化

光栅: 在德语中是屏幕的意思.

光栅化: 把东西画在屏幕上

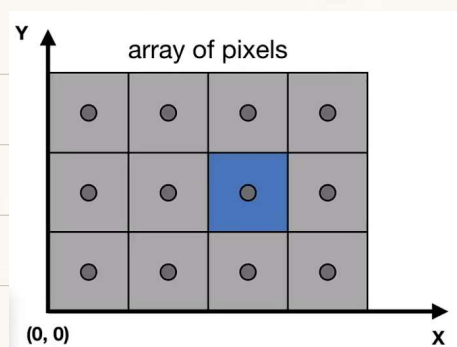
3. 像素

(1) 一个小的正方块, 有单独的颜色

(2) 颜色是红 绿 蓝 三色组成

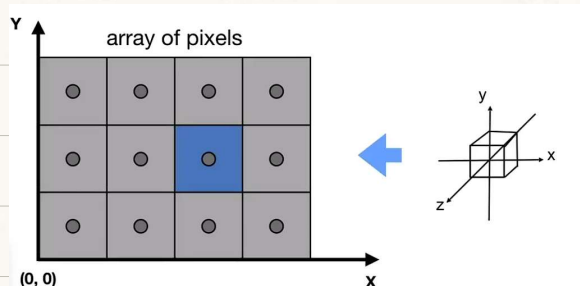
4. 屏幕空间

- (1) 像素使用 (x, y) 来描述位置
- (2) x, y 从 $(0, 0)$ 到 $(width-1, height-1)$, x, y 是整数
- (3) 像素 (x, y) 的中心是 $(x+0.5, y+0.5)$
- (4) 整个屏幕从 $(0, 0)$ 到 $(width, height)$



5. 从立方体到屏幕空间

- (1) 忽略 z
- (2) 将 x, y 从 $[-1, 1]^2$ 变换到 $[0, width] \times [0, height]$



(3) 变换矩阵 (视口变换)

$$M_{\text{viewpoint}} = \begin{bmatrix} \frac{width}{2} & 0 & 0 & \frac{width}{2} \\ 0 & \frac{height}{2} & 0 & \frac{height}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

三. 光栅化: 在屏幕上绘画

1. 三角形 —— 基本形状要素

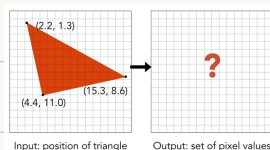
为什么是三角形

① 最基本的多边形: 任何多边形都能拆成三角形

② 独特的性质: 一定成为一个平面, 很好定义内部和外部

对三个顶点进行属性的定义可以很好地在三角形内部光滑地对属性进行变化

2. 从三角形到像素



四、简单的一种光栅化方法:采样

1. 基本原理:采样

```
for (int x=0; x<xmax; ++x) output[x] = f(x);
```

利用像素中心进行采样,判断像素中心是否在三角形内部

2. inside 函数

定义一个二元函数 inside

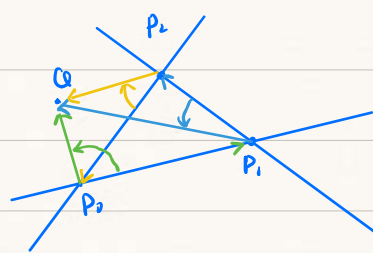
$$\text{inside}(t, x, y) = \begin{cases} 1 & \text{点}(x, y) \text{在} \\ & \text{三角形} t \text{内部} \\ 0 & \text{否则} \end{cases}$$

```
for (int x=0; x<xmax; ++x)
```

```
for (int y=0; y<ymax; ++y)
```

```
image[x][y] = inside(tri, x+0.5, y+0.5);
```

3. 如何判断一个点是否在三角形内部



$\vec{P_2P_0} \times \vec{P_2Q}$ 的方向指示 Q 在 P_2P_0 的右侧

$\vec{P_0P_1} \times \vec{P_0Q}$ 的方向指示 Q 在 P_0P_1 的左侧

$\vec{P_1P_2} \times \vec{P_1Q}$ 的方向指示 Q 在 P_1P_2 的左侧

} Q 在三角形外部

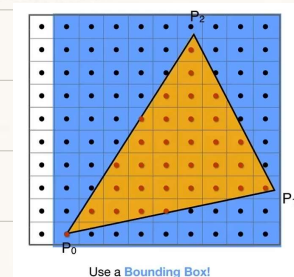
注意向量的顺序

PS: 在本门课上认为在边上的点在内部 在 OpenGL 中记上不计下, 记左不计右

4. 三角形的包围盒

对三角形的三个顶点求最大和最小值 (x 和 y) 形成矩形

不在包围盒中的像素可以不做判断



第六课 光栅化 (反走样和深度缓冲)

一、采样理论

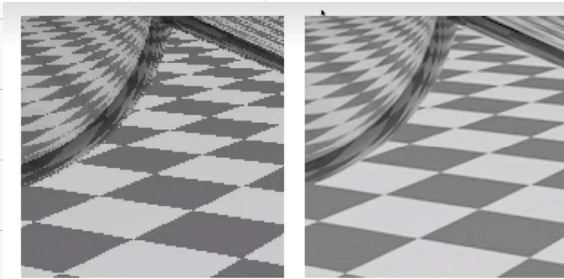
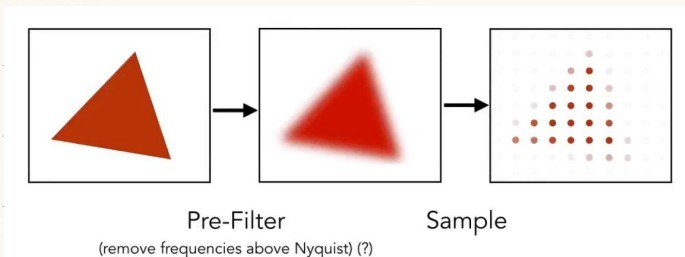
1. 3种采样中的错误

(1) 锯齿 (2) 摩尔纹 (3) 车轮效应

总结: 信号变化的速度太快 以至于采样速度跟不上

2. 如何反走样

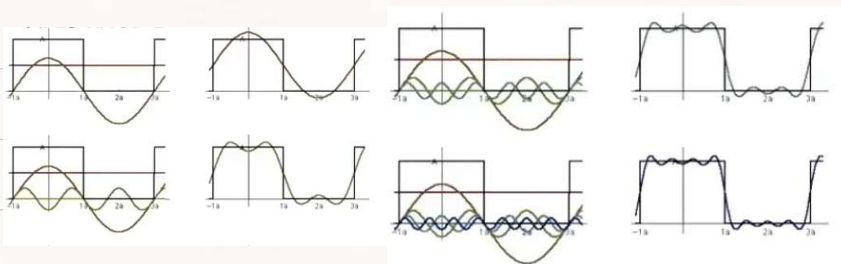
在采样前 先进行模糊操作, 然后再进行采样



3. 频域

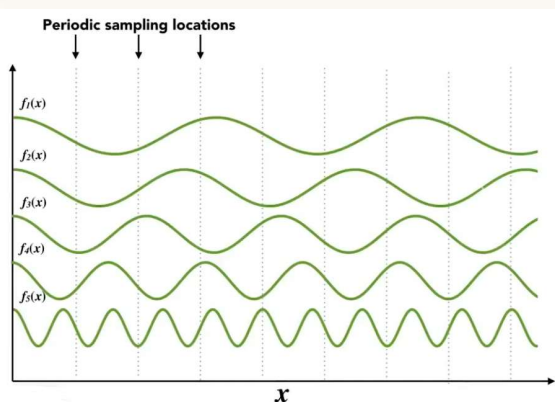
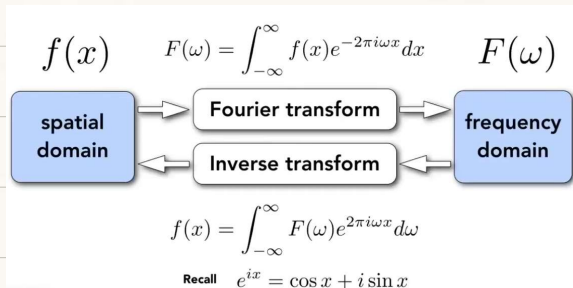
(1) 傅利叶展开

$$f(x) = \frac{A}{2} + \frac{2A \cos(1\pi x)}{\pi} - \frac{2A \cos(3\pi x)}{3\pi} + \frac{2A \cos(5\pi x)}{5\pi} - \frac{2A \cos(7\pi x)}{7\pi} + \dots$$

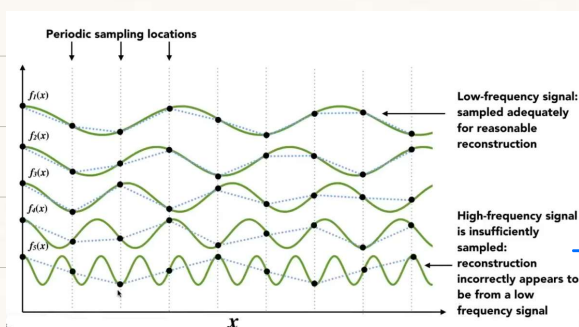


(2) 傅利叶变换

将函数 $f(x)$ 拆分为一系列不同频率的段



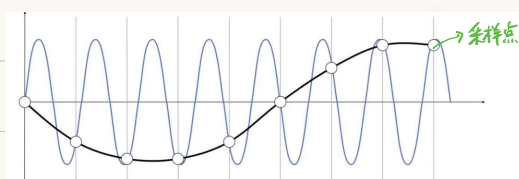
更高频率的信号需要更快的采样



→ 高频率下丢失较多信息

4. 走样的定义

定义: 使用同样的采样方法采样两种不同频率的函数, 得到的结果相同



图中蓝色曲线和黑色曲线的
采样结果相同

5. 滤波

(1) 定义: 删去一系列频率

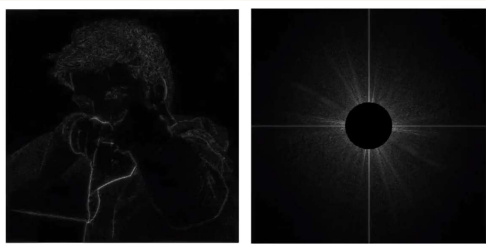


频率代表变化程度, 变化越剧烈

中心: 低频区域 周围: 高频区域

水平和 竖直: 四方连续 (将图片平铺) 的结果

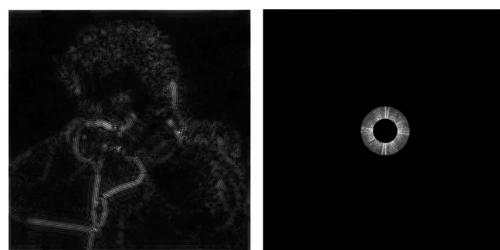
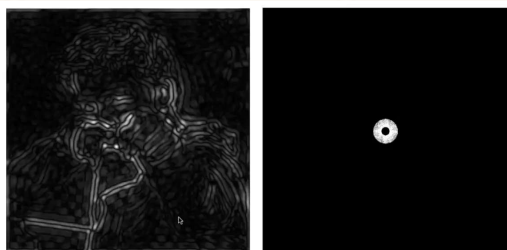
(1) 高通滤波 只通过高频信号



(3) 低通滤波 只通过低频信号



(4) 带通滤波 通过中间某一带的信号.

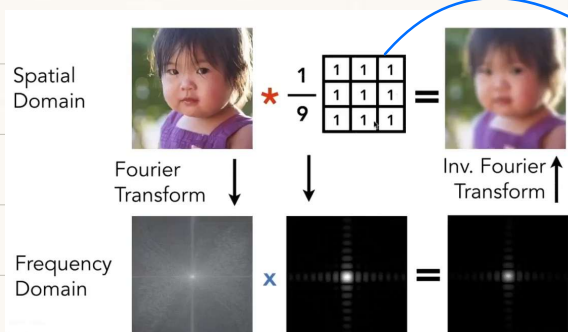


滤波 = 平均 = 卷积

(5) 卷积 对中心像素、周围像素求加权平均值

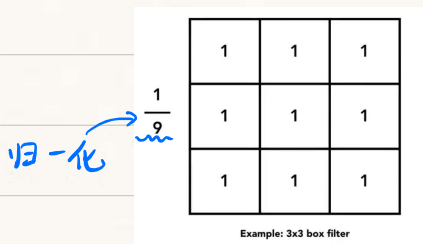
Signal	1	3	5	3	7	1	3	8	6	4
Filter	1/4	1/2	1/4							
	$3 \times (1/4) + 5 \times (1/2) + 3 \times (1/4) = 4$									
Result		3	4							

卷积的定理: 对两个信号在时域上的卷积相当于对两个信号在频域上的乘积



→ 任何一个像素是它周围3x3像素的平均值

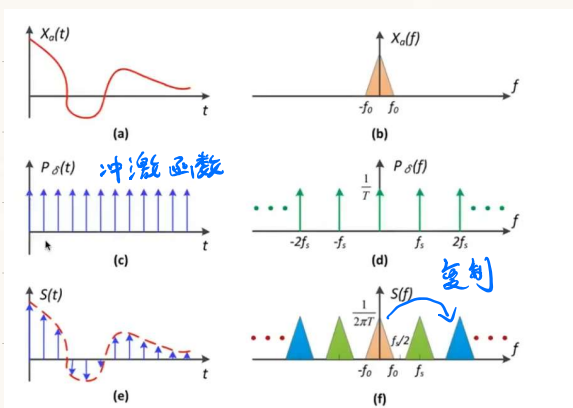
1.6) box filter



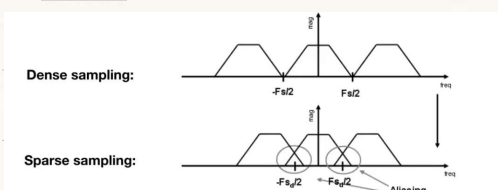
box 越大, 留下的频率越低, 图片越模糊

6. 采样

1) 采样 重复频率内容



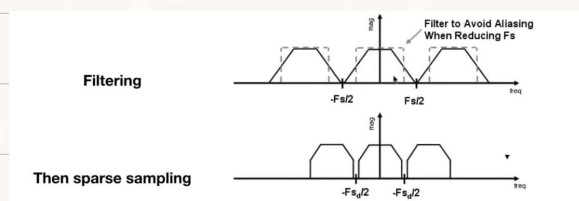
2) 走样 重复频率内容时发生了重叠



3) 反走样

① 增加采样率

② 反走样: 模糊后采样, 去除高频信号再采样

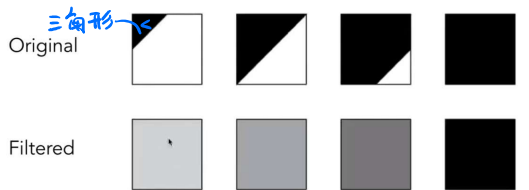


7. 如何模糊

(1) 卷积 使用一个像素大小的 box filter

(2) 取样 对每个像素中心取样

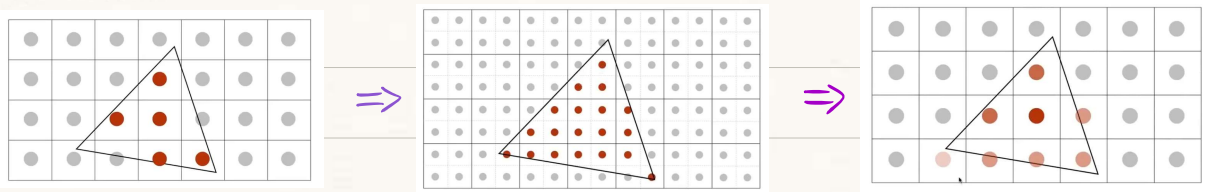
在光栅化一个三角形时, 一个像素中的平均值是像素中三角形覆盖的大小.



二. 超采样反走样(MSAA)

1. 工作原理(执行模糊)(没有提高采样率)

将一个像素划分为更小的区域, 对于这些更小的区域求平均值



2. MSAA 的性能开销

用了更多子像素进行判断

3. FXAA (Fast Approximate AA)

检测图像中的边界 \Rightarrow 将有锯齿的边界替换为无锯齿的边界

4. TAA (Temporal AA)

将MSAA的样本的历史样本信息应用于这一帧

5. 超分辨率/超采样

低分辨率图 \Rightarrow 高分辨率 (DLSS) 深度学习