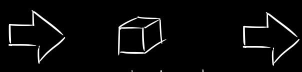
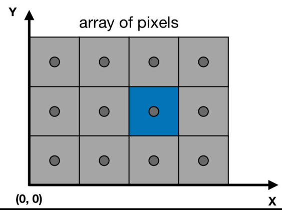
Orthographic
Perspective



canonical cube ? 多边形转化为屏幕上的像素值

- What is a screen?
  - An array of pixels
  - Size of the array: resolution
  - A typical kind of raster display
- Raster == screen in German
  - Rasterize == drawing onto the screen
- Pixel (FYI, short for "picture element")
  - For now: A pixel is a little square with uniform color
  - Color is a mixture of (red, green, blue)

- Defining the screen space
  - Slightly different from the "tiger book"



Pixels' indices are in the form of (x, y), where both x and y are integers

Pixels' indices are from (0, 0) to (width - 1, height - 1)

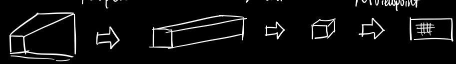
Pixel (x, y) is centered at (x + 0.5, y + 0.5)

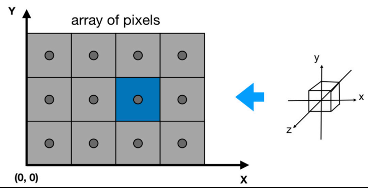
The screen covers range (0, 0) to (width, height)

Defining Screen Space

Canonical Cube to Screen

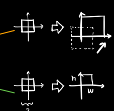
$M_{pers-ortho}$ M_{ortho} $M_{viewpoint}$





- irrelevant to z
- Transform in xy plane : $[-1, 1]^2$ to $[0, width] \times [0, height]$
- Viewport transform matrix

$$M_{viewport} = \begin{pmatrix} \frac{width}{2} & 0 & 0 & \frac{width}{2} \\ 0 & \frac{height}{2} & 0 & \frac{height}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



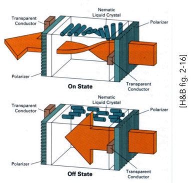
LCD

LCD (Liquid Crystal Display) Pixel

Principle: block or transmit light by twisting polarization

Illumination from backlight (e.g. fluorescent or LED)

Intermediate intensity levels by partial twist



[H&B fig. 2-16]

液晶扭曲造成光线偏振
以显示该像素内容

Rasterization

Drawing to raster display

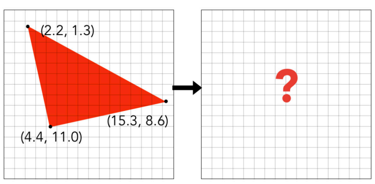
Meshes { polygon  / triangle 

Why triangles ?

- Most basic polygon → can break up other shapes
- Unique properties
  → guaranteed to be planer
  → well-defined interior
  → well-defined method for interpolating values at vertices over triangle
  barycentric interpolation

Triangle

What Pixel Values Approximate a Triangle?



Input: position of triangle vertices projected on screen

Output: set of pixel values approximating triangle

Sampling

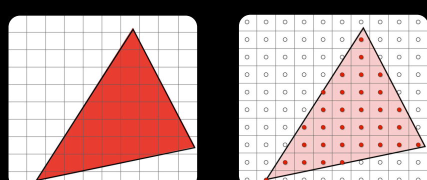
a simple approach 利用像素中心对屏幕空间采样

Evaluating a function at a point is sampling.

We can discretize a function by sampling.

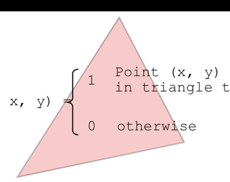
```
for (int x = 0; x < xmax; ++x)
    output[x] = f(x);
```

Sampling is a core idea in graphics.
We sample time (1D), area (2D), direction (2D), volume (3D) ...



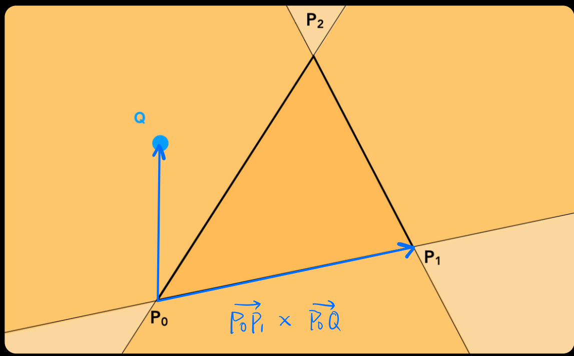
定义函数

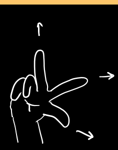
判断像素中心
是否在图形内



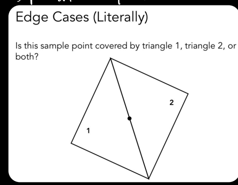
$$\text{inside(t, x, y)} = \begin{cases} 1 & \text{Point (x, y) in triangle t} \\ 0 & \text{otherwise} \end{cases}$$

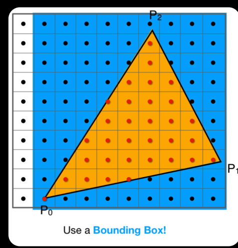
```
for (int x = 0; x < xmax; ++x)
  for (int y = 0; y < ymax; ++y)
    image[x][y] = inside(tri,
                         x + 0.5,
                         y + 0.5);
```



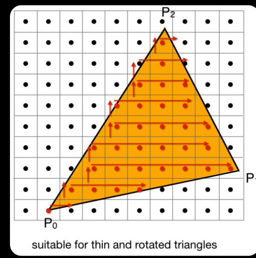


本课选择忽略

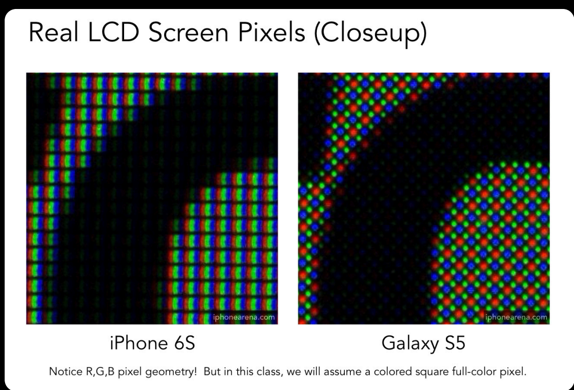




Axis Aligned Bounding Box
AABB



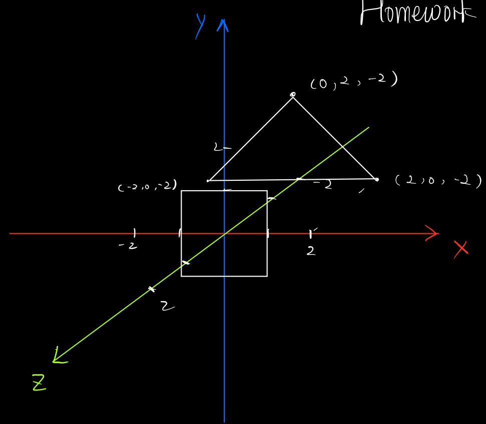
Incremental triangle traversal



RGBG

绿色密度最大 → 绿色对于人眼更加敏感

Homework 1 Note

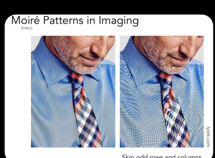


Antialiasing

抗锯齿（反走样）

Sampling is ubiquitous in computer graphics

Sampling artifact < error / mistakes / inaccuracies >



Artifacts due to sampling - "Aliasing"

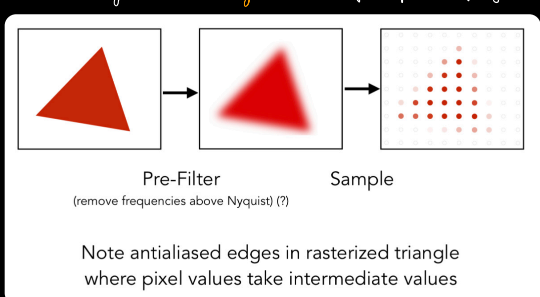
- Jaggies – sampling in space
- Moire – undersampling images
- Wagon wheel effect – sampling in time
- [Many more] …

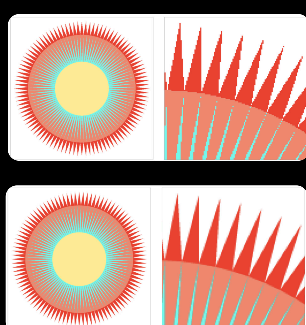
Behind the Aliasing Artifacts

- Signals are changing too fast (high frequency), but sampled too slowly

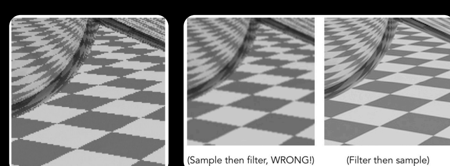
☆

Antialiasing idea: blurring < pre-filtering > before sampling





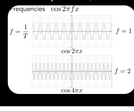
BLURRED ALIASING



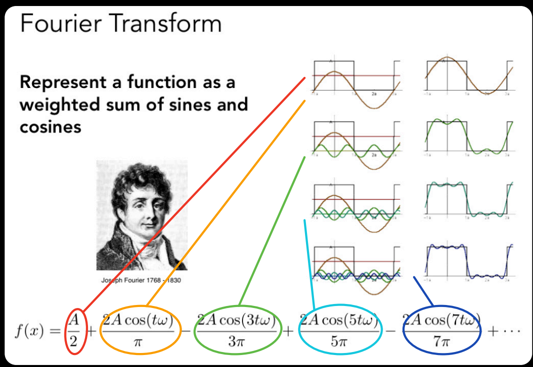
Filter + Sample

Fourier Transform 傅里叶级数展开与傅里叶变换

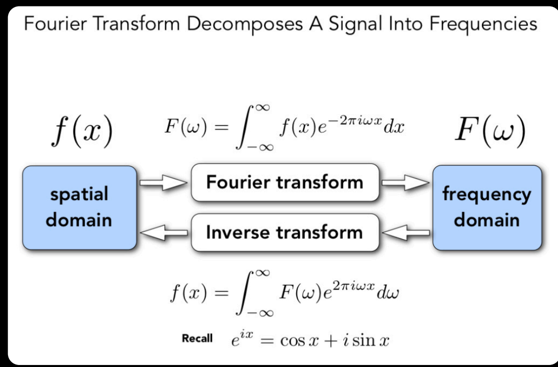
频率与频域



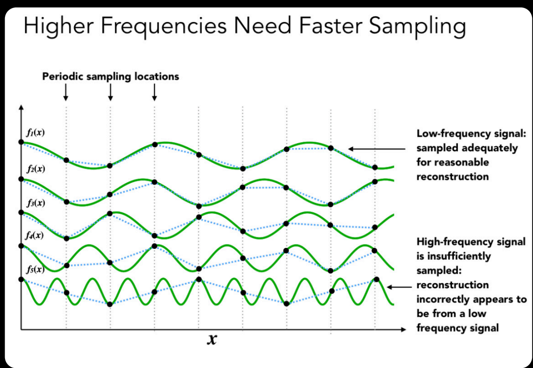
周期 = $\frac{1}{频率}$



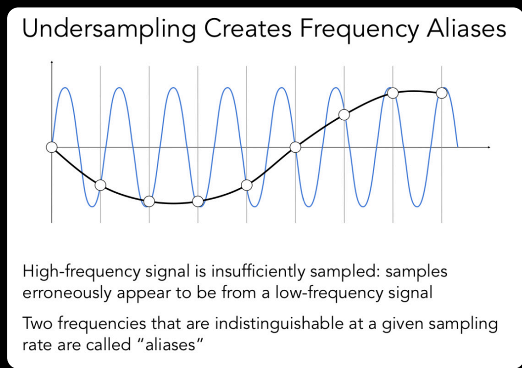
频率递增 (↑)



通过存储频率以之后复原

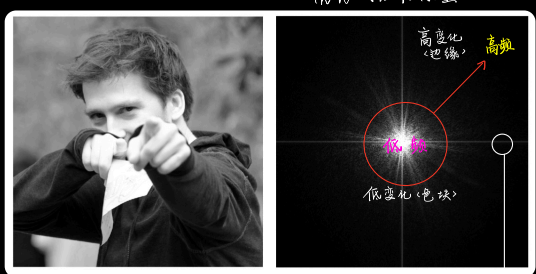


高频率函数难以精确采样 ⟶ 类似于走样



Filtering = Getting rid of certain frequency contents

亮度代表信息量

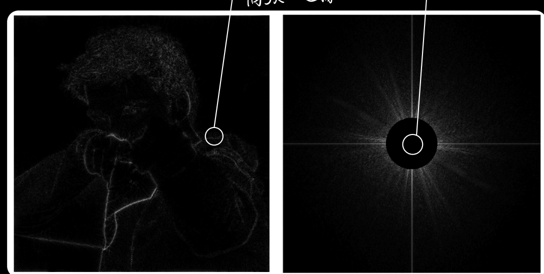


时域 ⇨ 频率

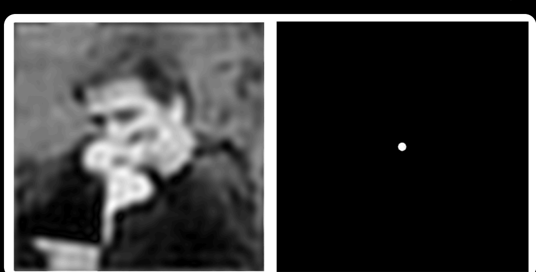
图像边界之间完全不相似 因而产生剧烈频率

突出剧烈变化的 高频→边缘

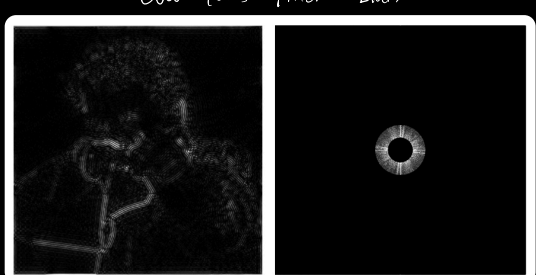
滤除低频波 以突出边缘

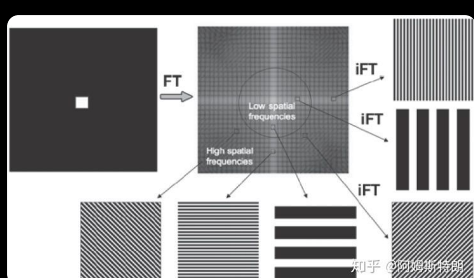


High Pass Filter

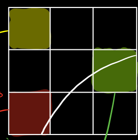


Low Pass Filter <blur>





天明显变化，低



明显变化，高

略有变化，中

原图

高频率〈轮廓〉
在图中的出场率

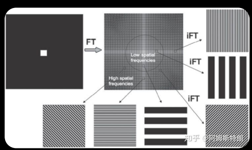
///

低频率〈色块〉
在图中的出场率

|||

\\\

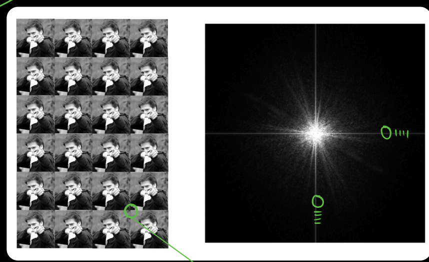
\\\ 频率越高，越具"方向性"



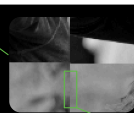
高频
细轮廓

低频
粗轮廓

关于"两条白线"



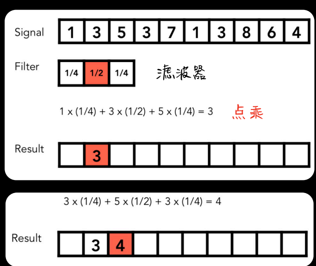
反复采样中…



高强度反复出现
"水平"与"竖直"的
高频率信号〈轮廓〉

也有"渐变"，因此这种"水平"/"竖直"
信号会遍布低/高频

Convolution 卷积



滤波器

点乘

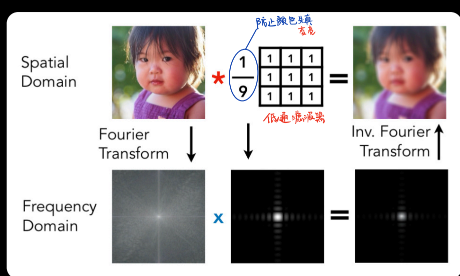
Convolution in the spatial domain is equal to multiplication in the frequency domain, and vice versa

Option 1:

- Filter by convolution in the spatial domain

Option 2:

- Transform to frequency domain (Fourier transform)
- Multiply by Fourier transform of convolution kernel
- Transform back to spatial domain (inverse Fourier)



防止颜色填充亮

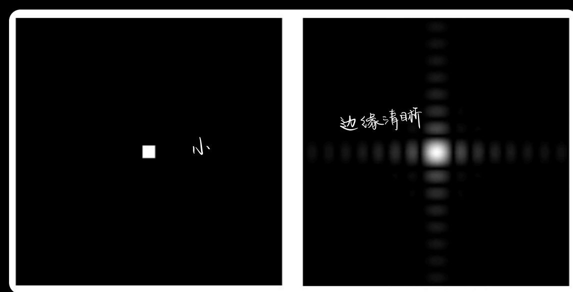
低通滤波器

只剩低频

时域上的卷积 = 频域上的乘积

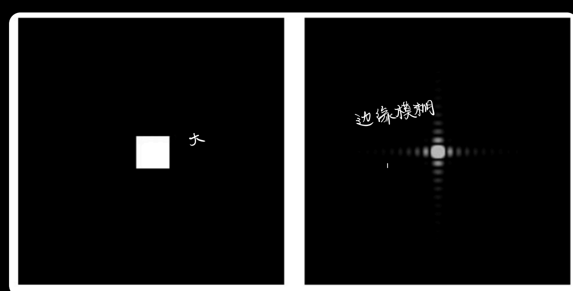
Filtering = Convolution 卷积

= Averaging 平均〈模糊〉



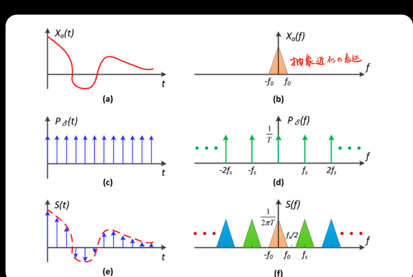
小

边缘清晰



大

边缘模糊



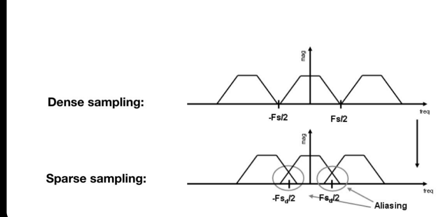
抽象近似の表达

(b) 卷积 (d) = (f)

信号(a)乘以冲激函数
得到采样(多个点)

频域中的频率与时域中的间隔为倒数关系，因此时域
中采样间隔过大，在频域中会体现为↓

Aliasing = Mixed Frequency Contents



产生走样

Sampling = Repeating Frequency Contents