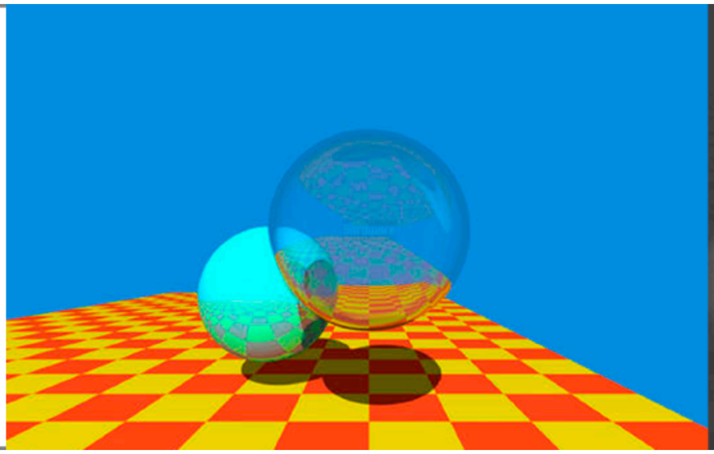
Introduction to Computer Graphics

GAMES101, Lingqi Yan, UC Santa Barbara

Lecture 6:
Rasterization 2
(Antialiasing and Z-Buffering)






http://www.cs.ucsb.edu/~lingqi/teaching/games101.html

Announcements

- Homework 1
  - Already 49 submissions so far!
  - In general, start early
- Today's topics are not easy
  - Having knowledge on Signal Processing is appreciated
  - But no worries if you don't

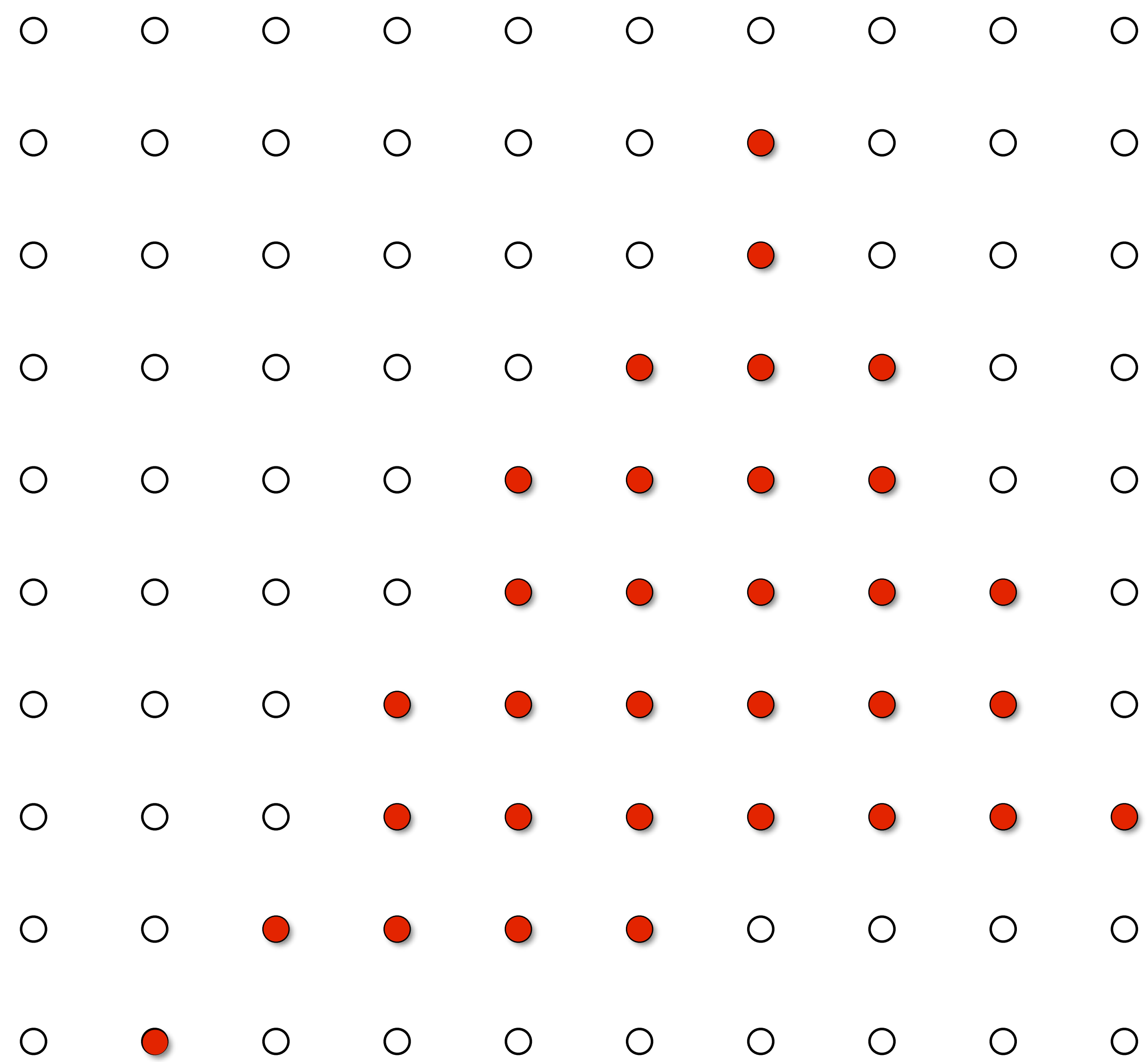
Last Lectures

- Viewing
  - View + Projection + Viewport
- Rasterizing triangles
  - Point-in-triangle test
  - Aliasing

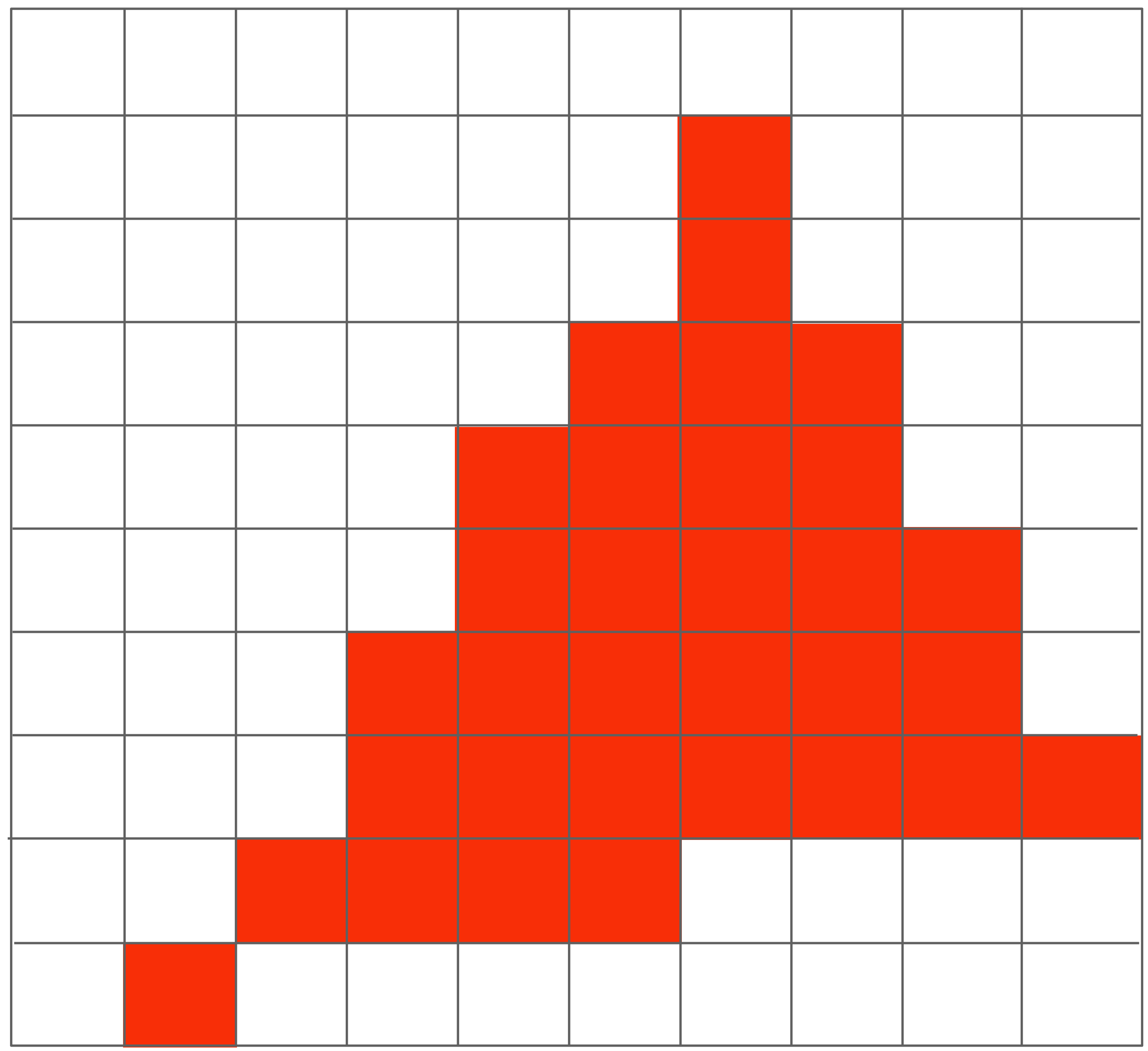
Today

- Antialiasing
  - Sampling theory
  - Antialiasing in practice
- Visibility / occlusion
  - Z-buffering

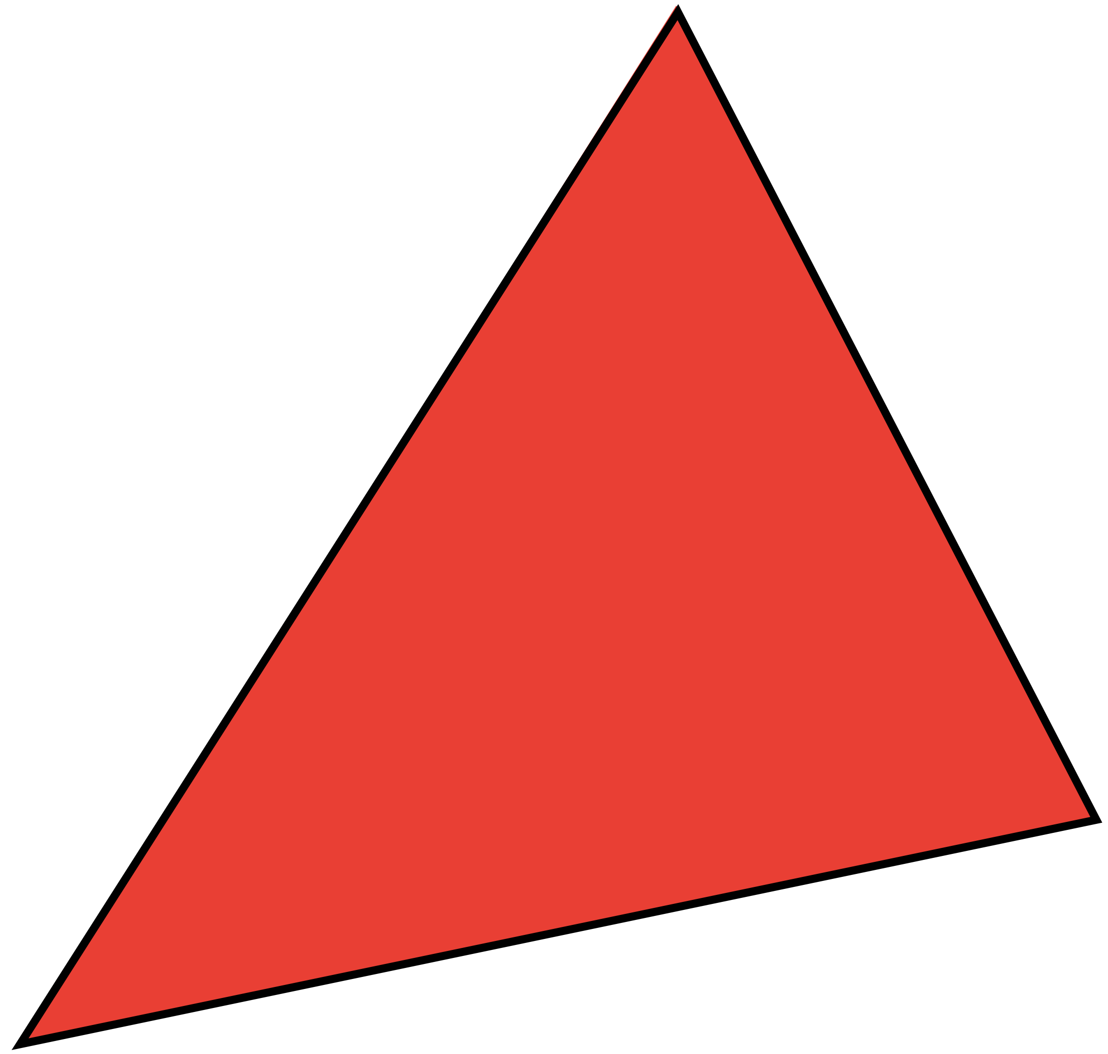
Recap: Testing in/out △ at pixels' centers



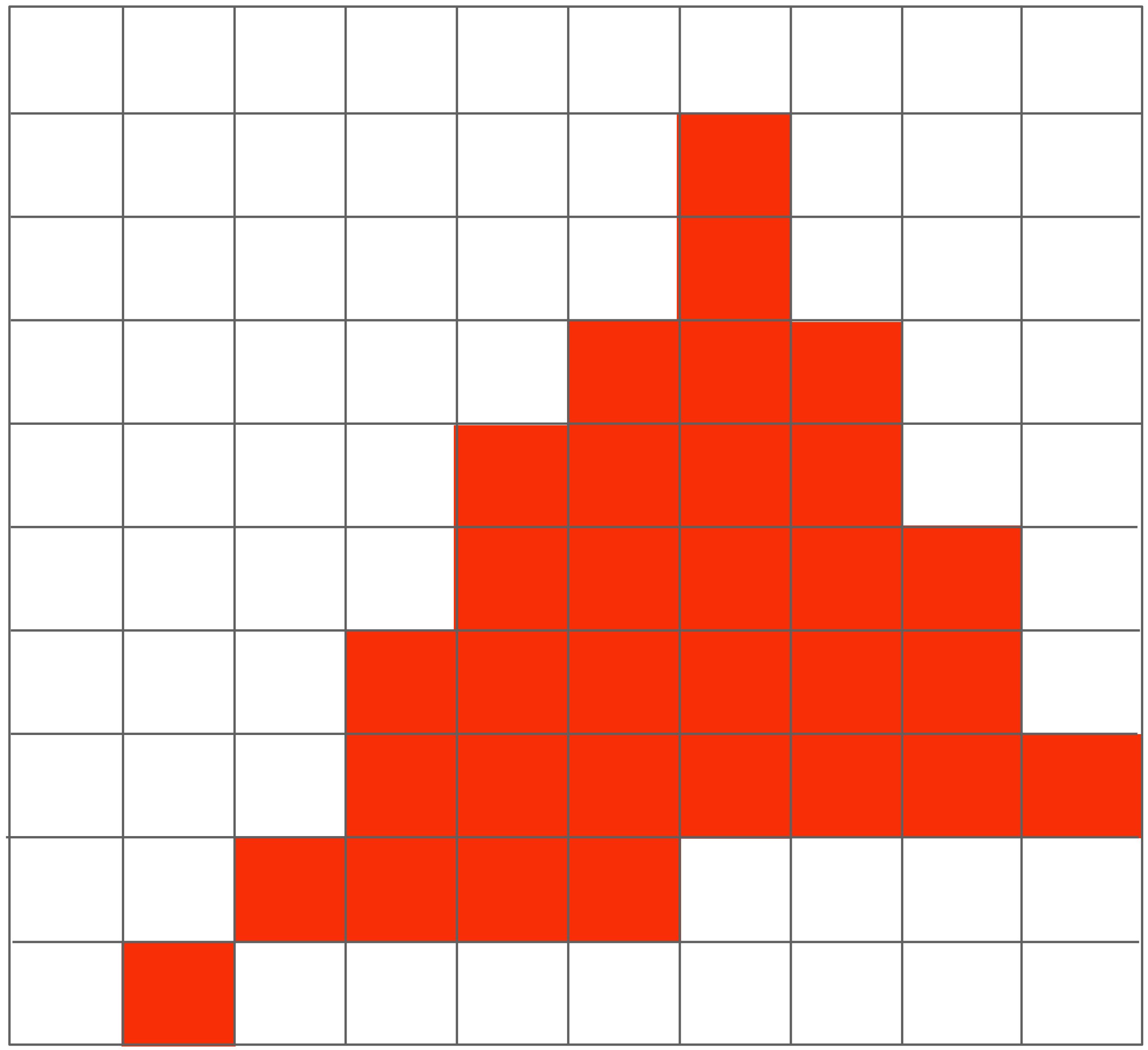
Pixels are uniformly-colored squares



Compare: The Continuous Triangle Function



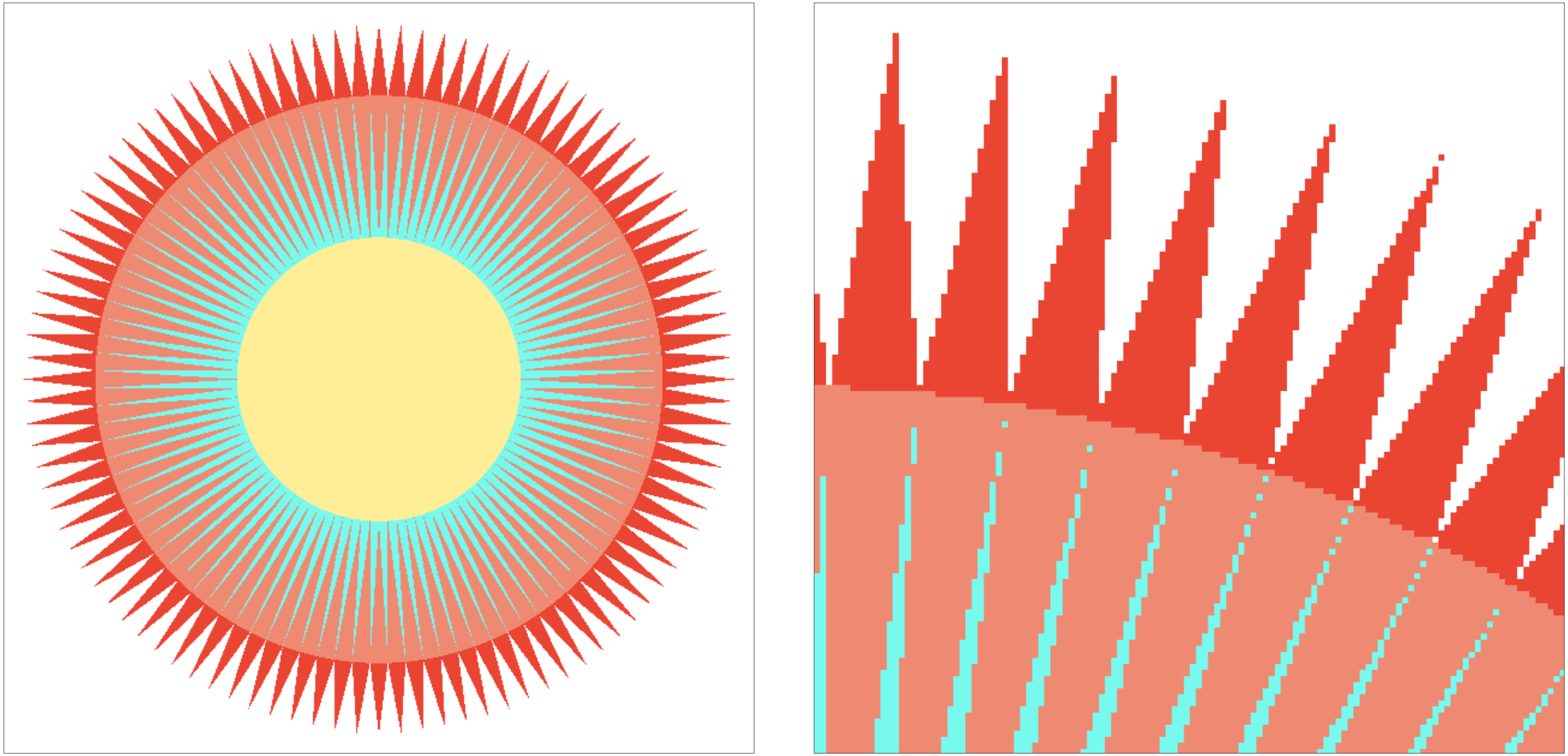
What’s Wrong With This Picture?



Jaggies!

Aliasing

走样
锯齿

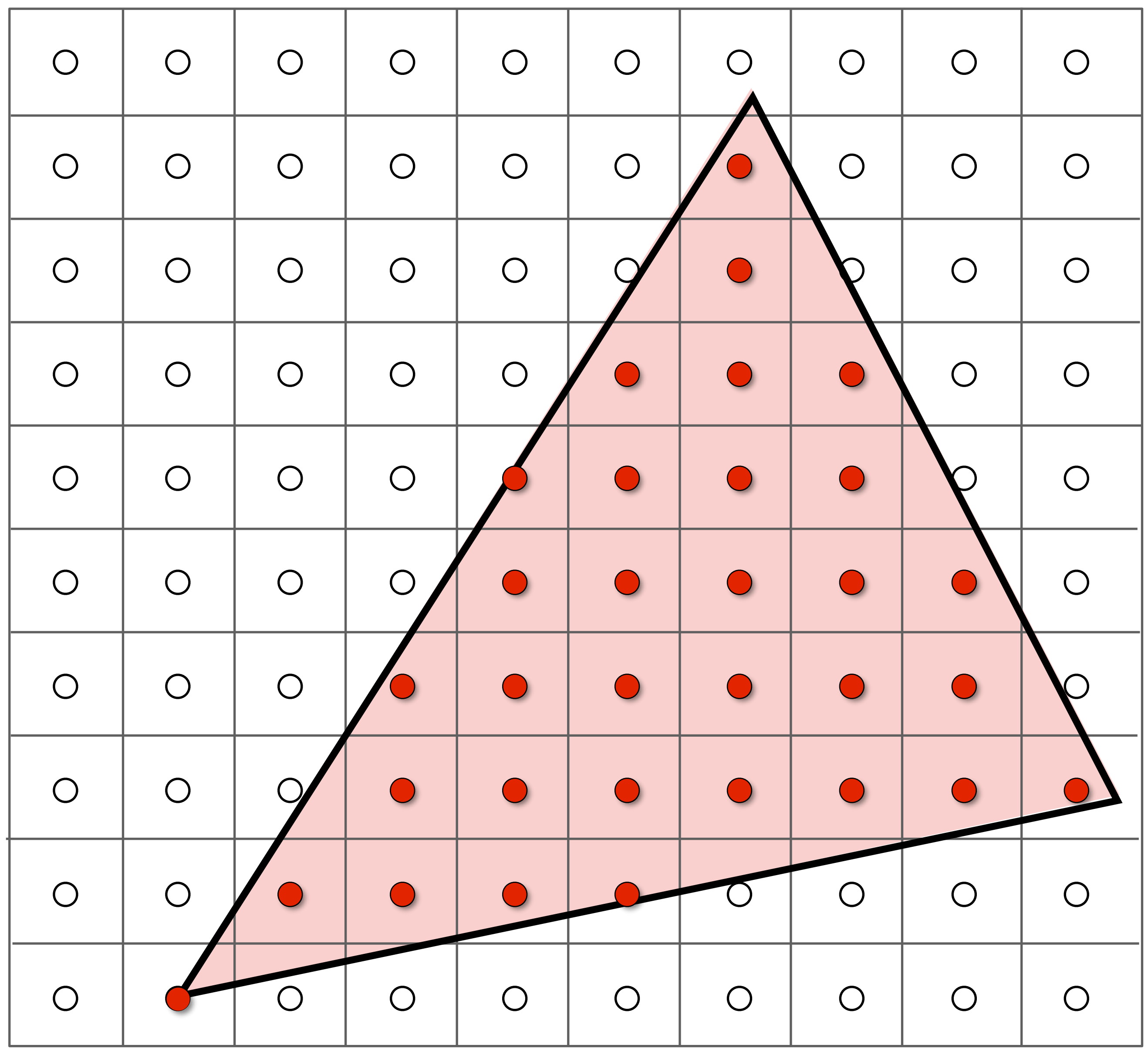


Is this the best we can do?

Slide courtesy of Prof. Ren Ng, UC Berkeley

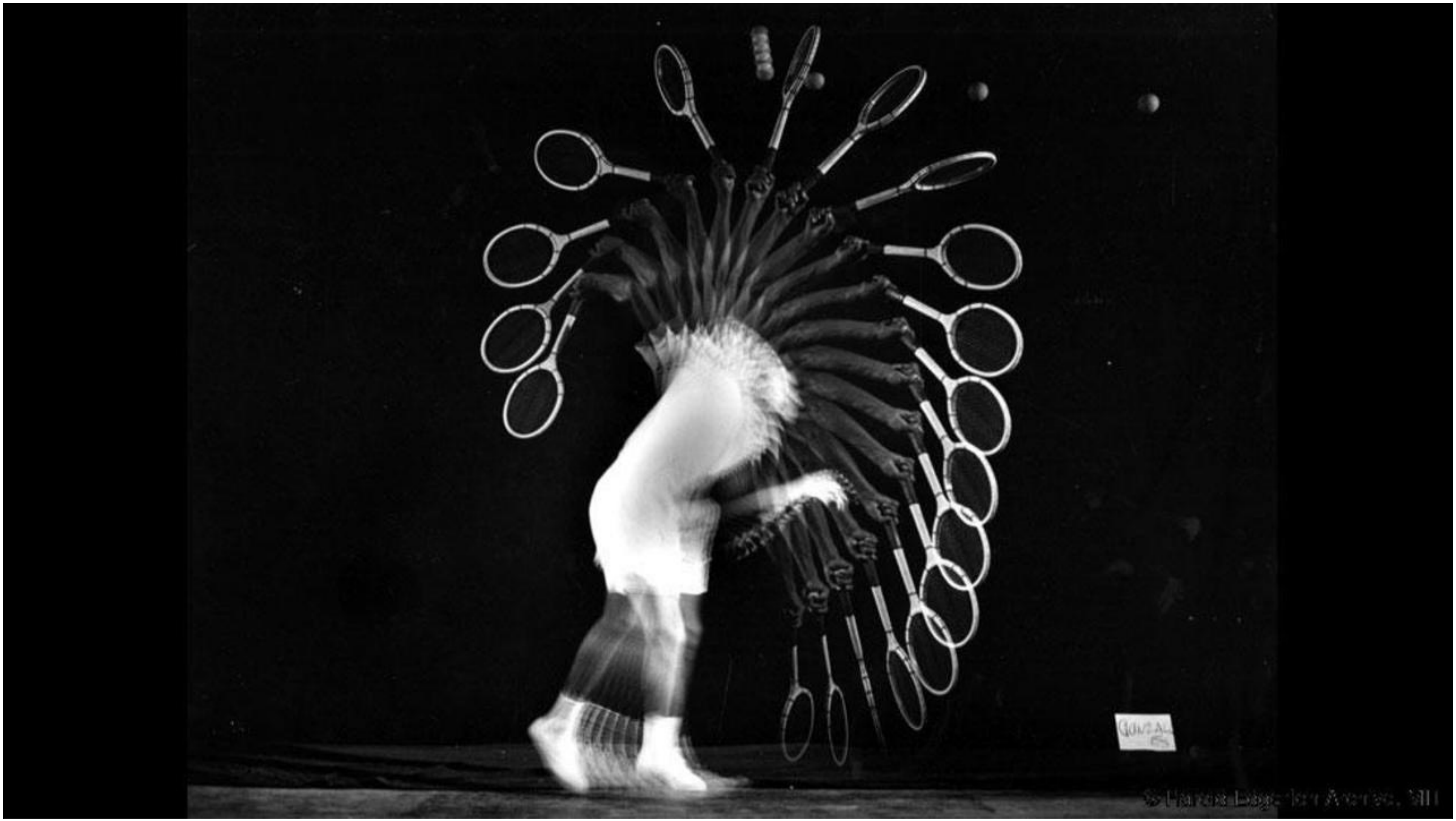
Sampling is Ubiquitous in Computer Graphics

Rasterization = Sample 2D Positions



Photograph = Sample Image Sensor Plane

Video = Sample Time

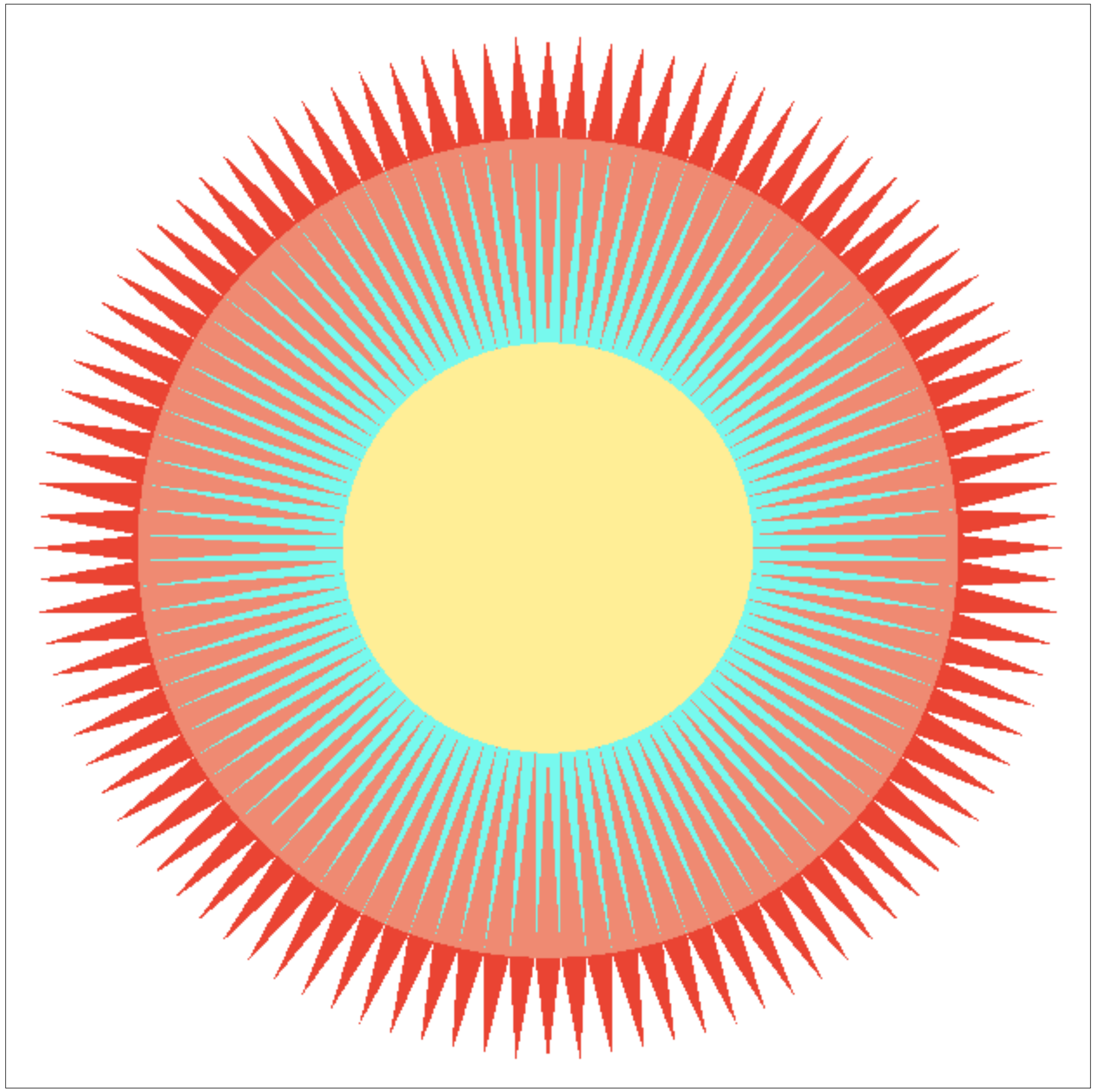


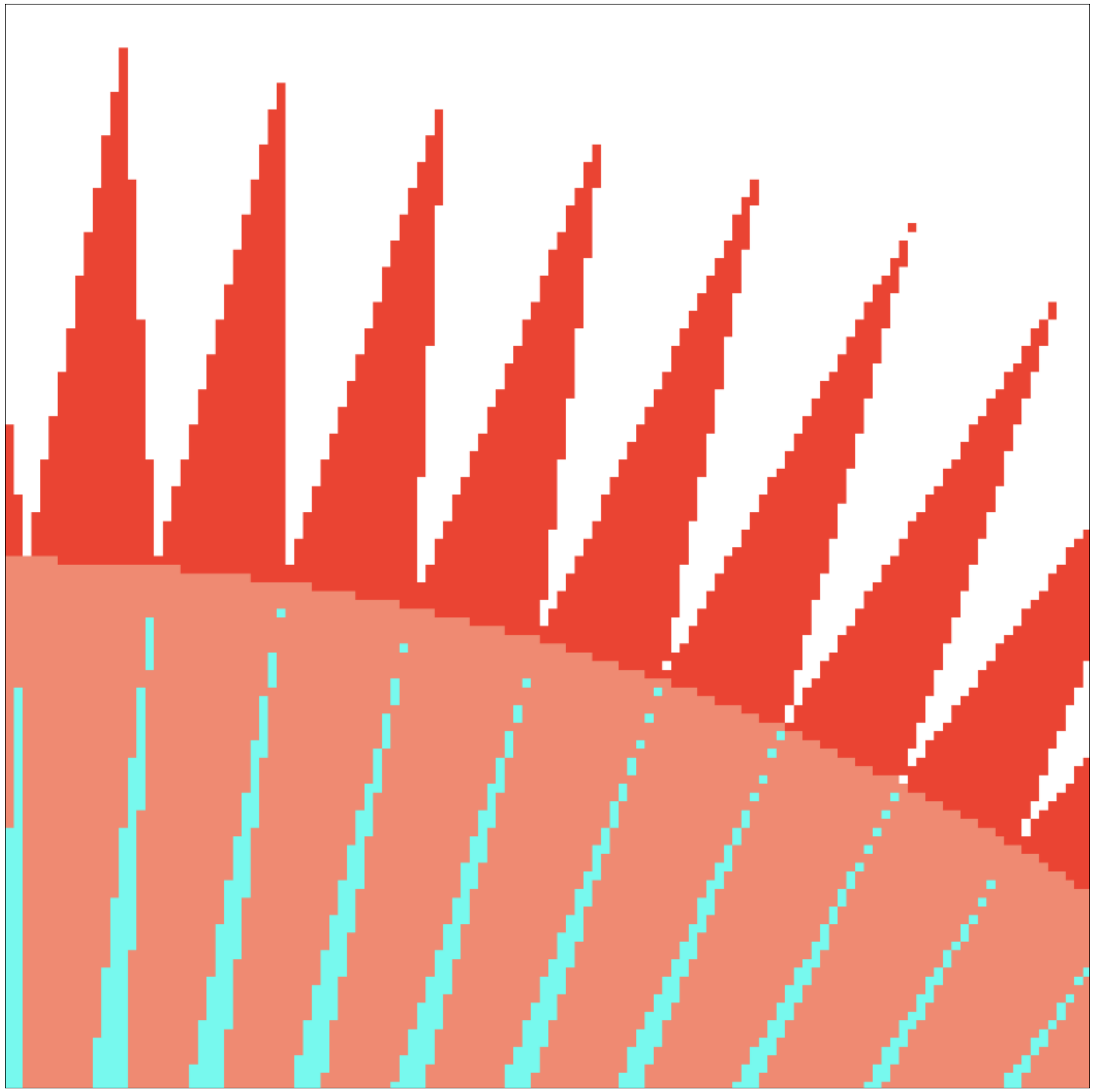
Harold Edgerton Archive, MIT

瑕疵

Sampling Artifacts
(Errors / Mistakes / Inaccuracies) in Computer Graphics

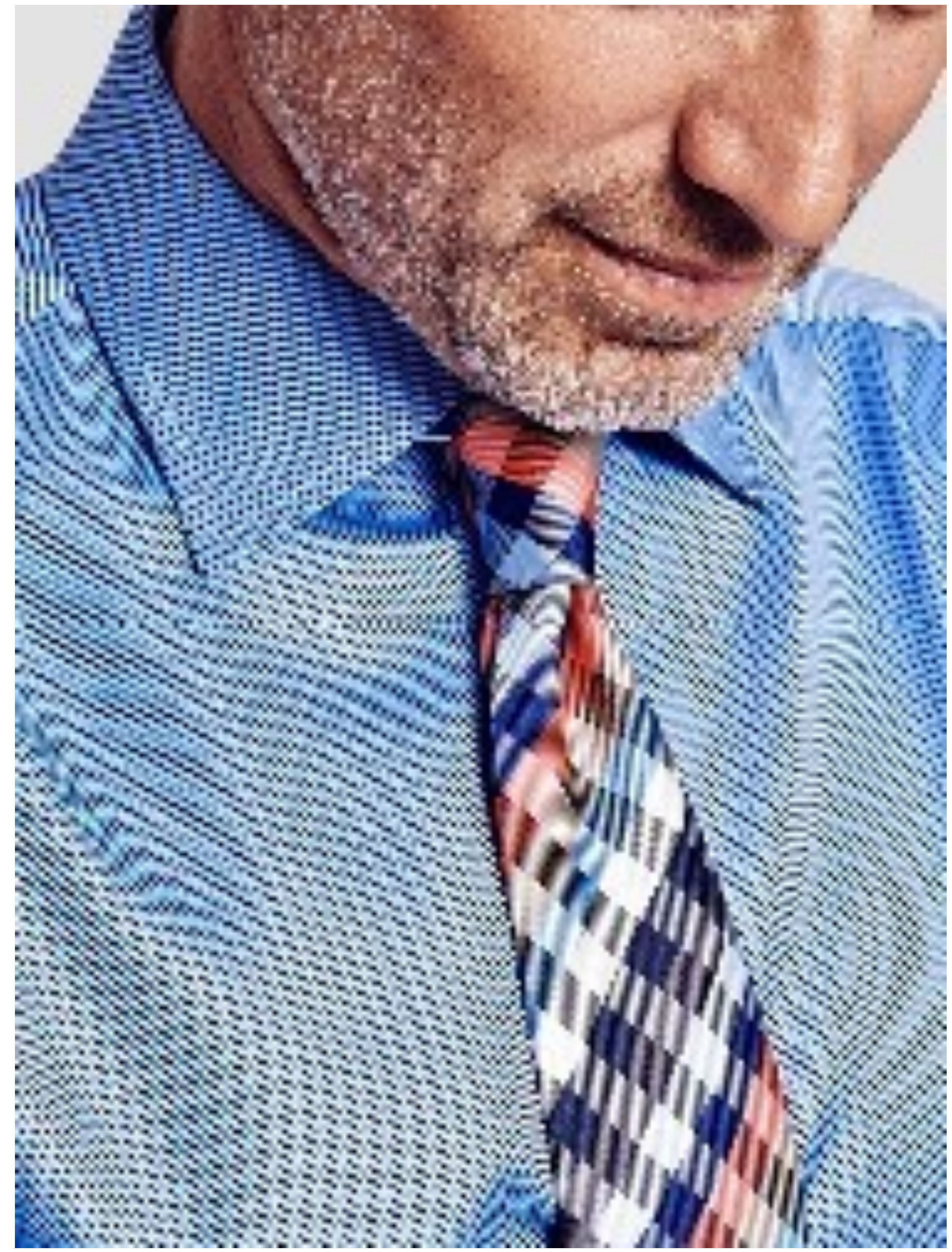
Jaggies (Staircase Pattern)





This is also an example of “aliasing” – a sampling error

Moiré Patterns in Imaging
[mwɑ:]



lystit.com

Skip odd rows and columns

Wagon Wheel Illusion (False Motion)

Sampling Artifacts in Computer Graphics

Artifacts due to sampling - “Aliasing”

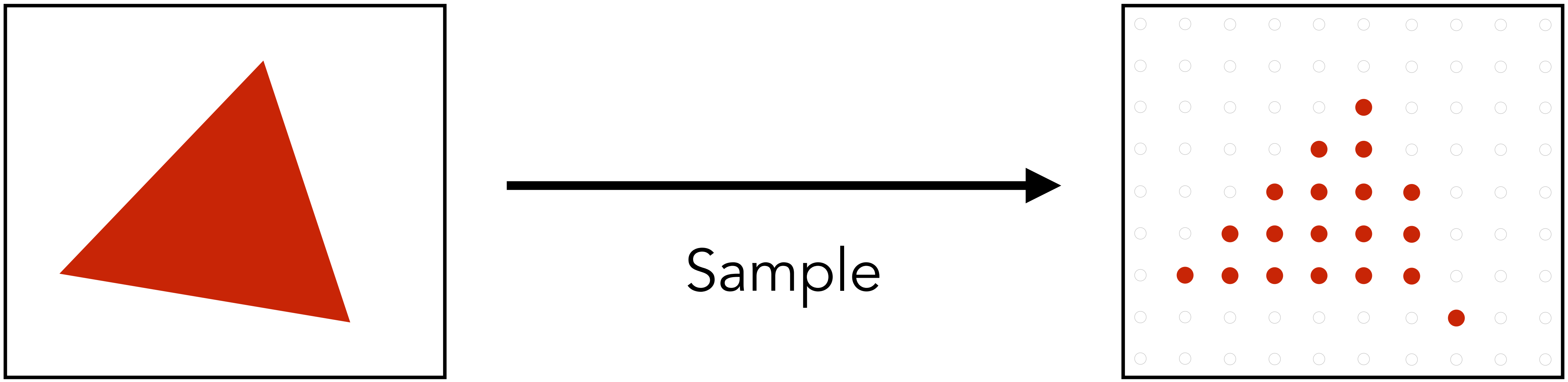
- Jaggies – sampling in space
- Moire – undersampling images
- Wagon wheel effect – sampling in time
- [Many more] …

Behind the Aliasing Artifacts

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

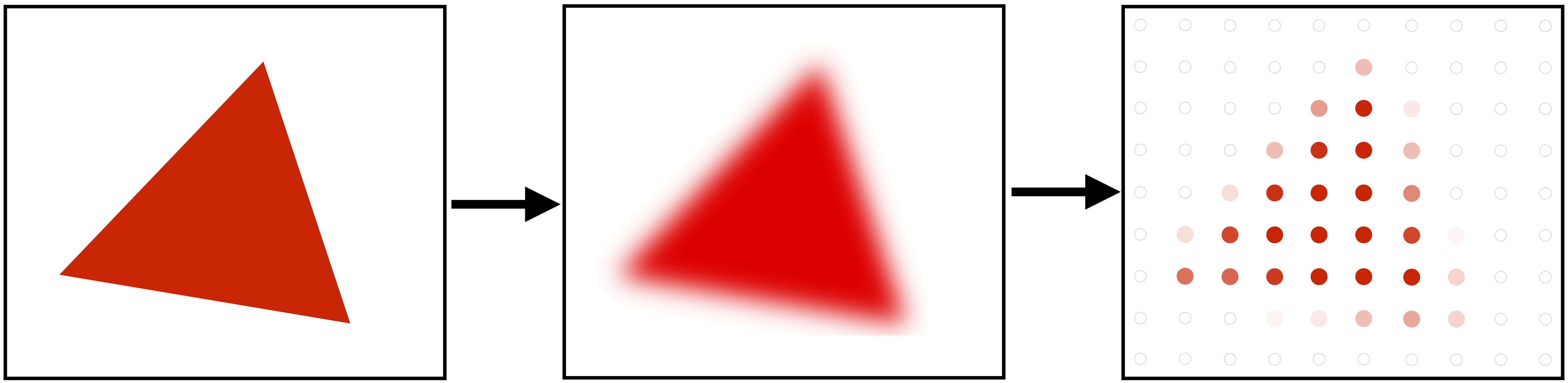
Antialiasing Idea:
Blurring (Pre-Filtering) Before Sampling

Rasterization: Point Sampling in Space



Note jaggies in rasterized triangle
where pixel values are **pure red or white**

Rasterization: Antialiased Sampling



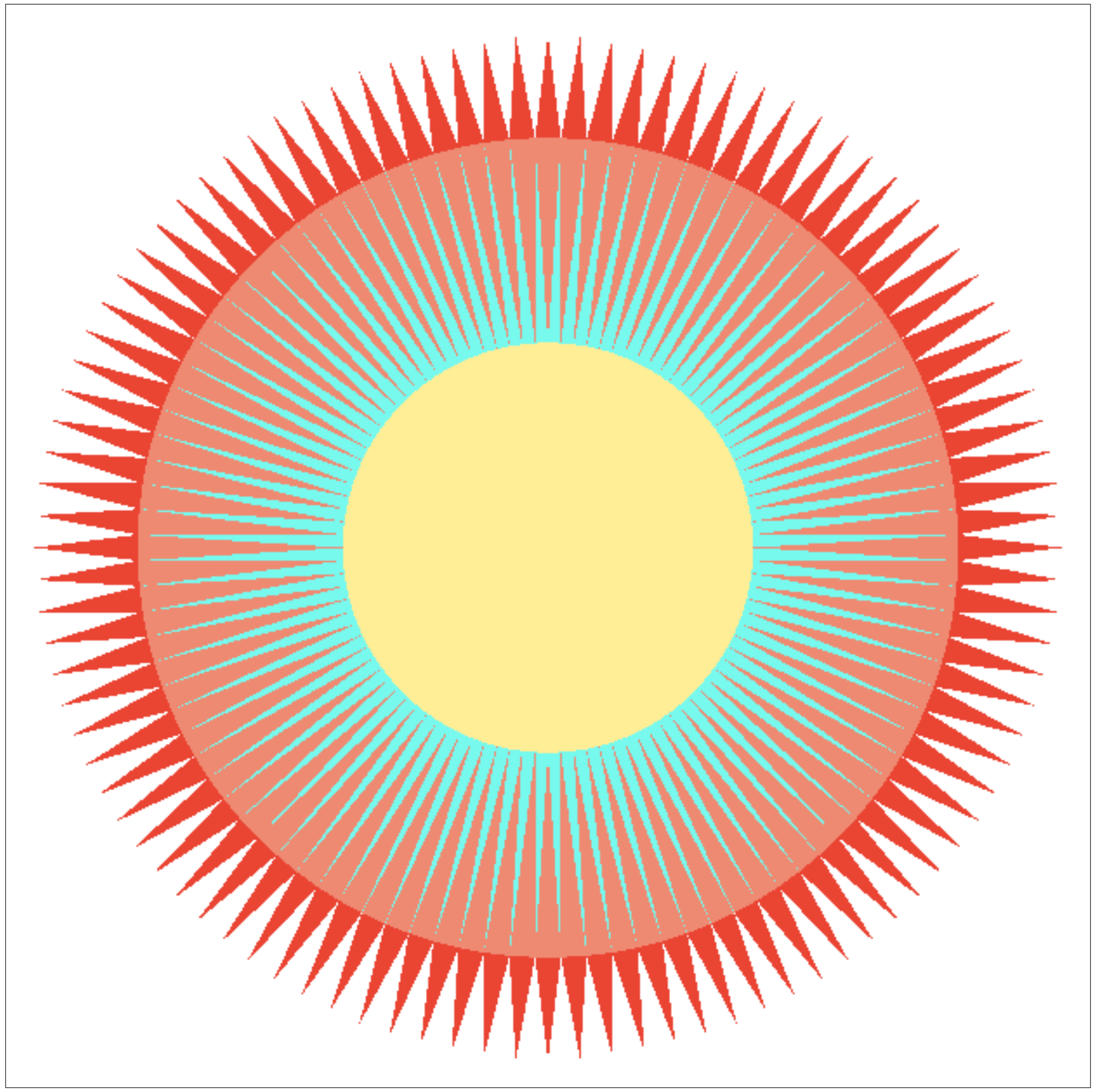
Pre-Filter

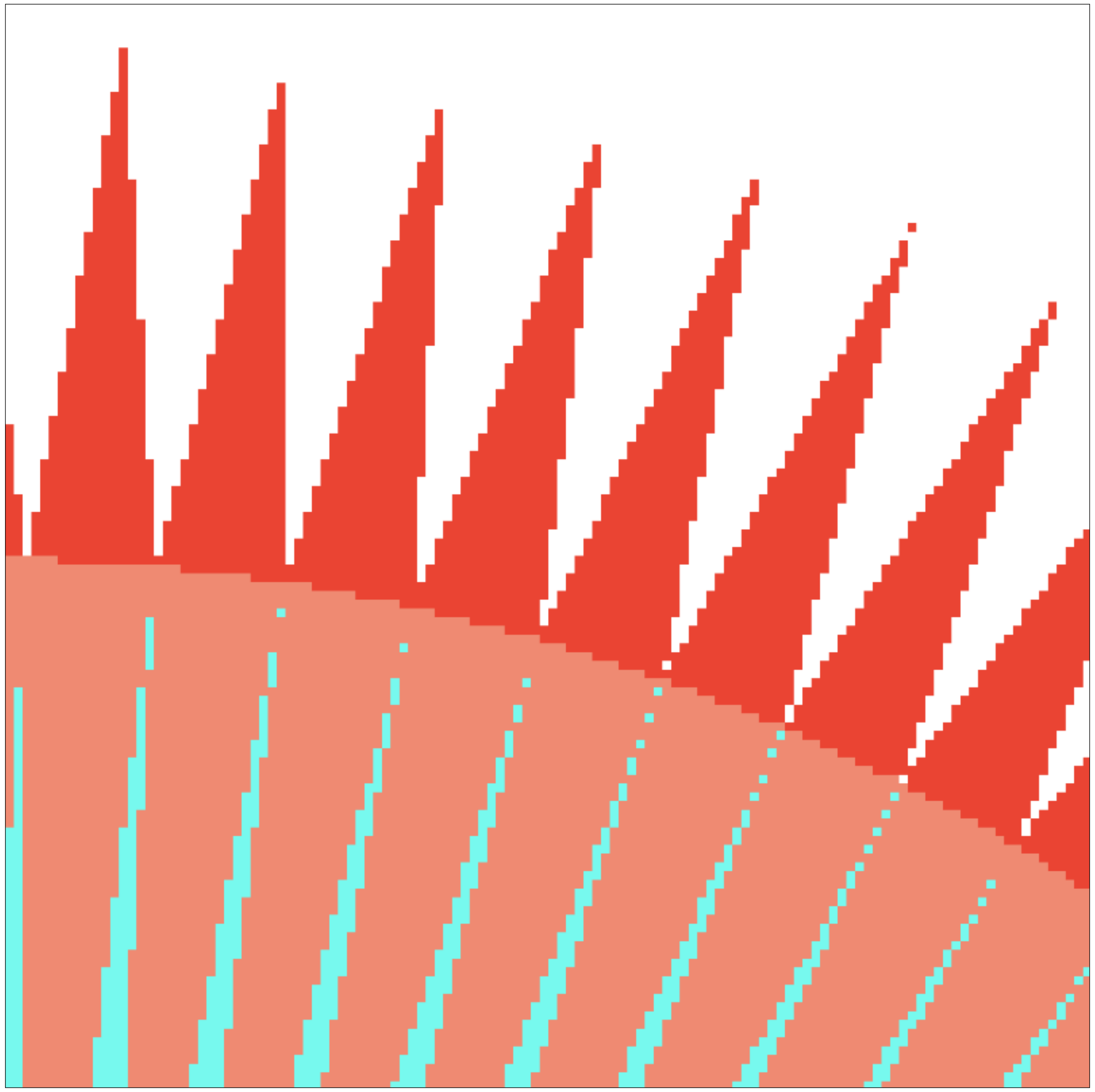
(remove frequencies above Nyquist) (?)

Sample

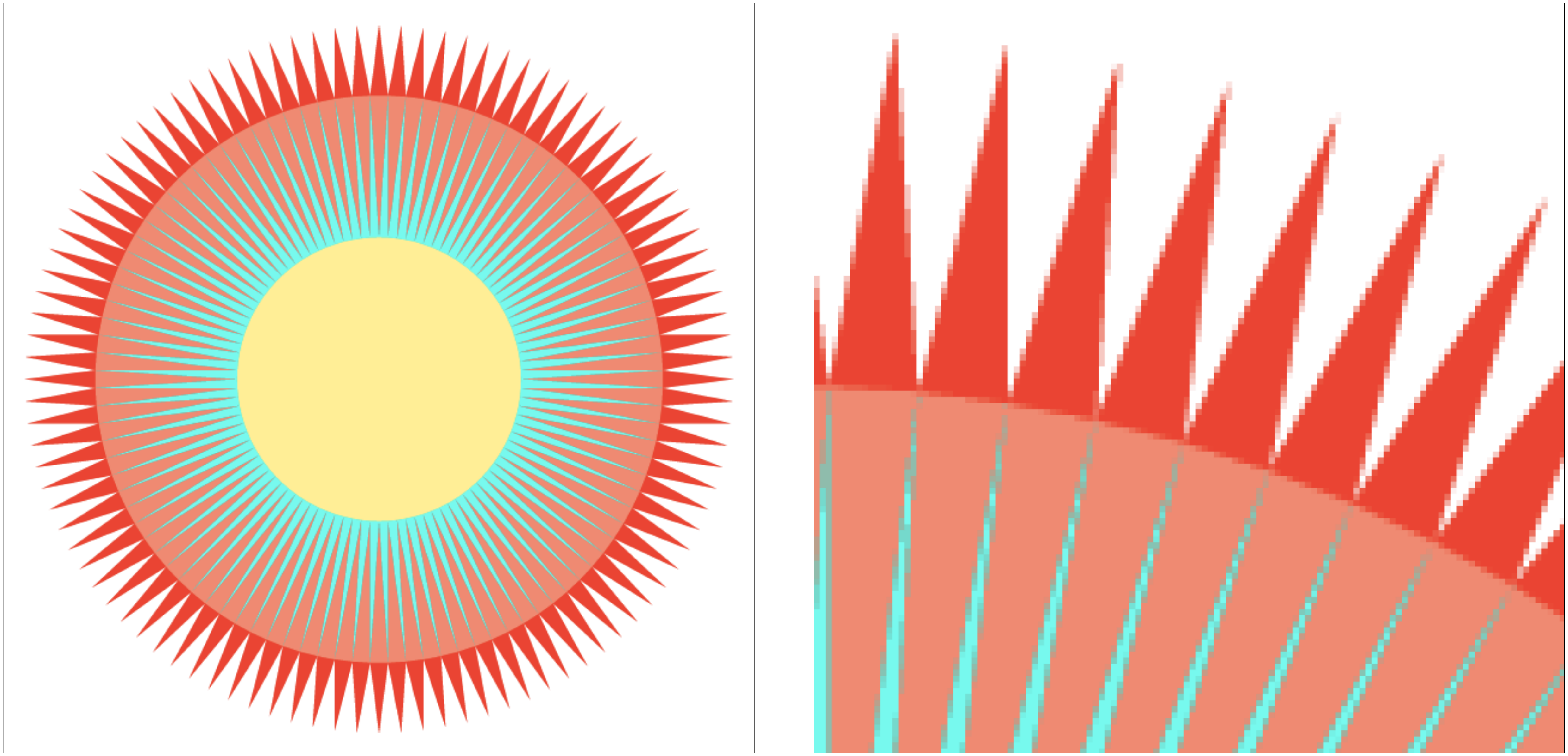
Note antialiased edges in rasterized triangle where pixel values take intermediate values

Point Sampling

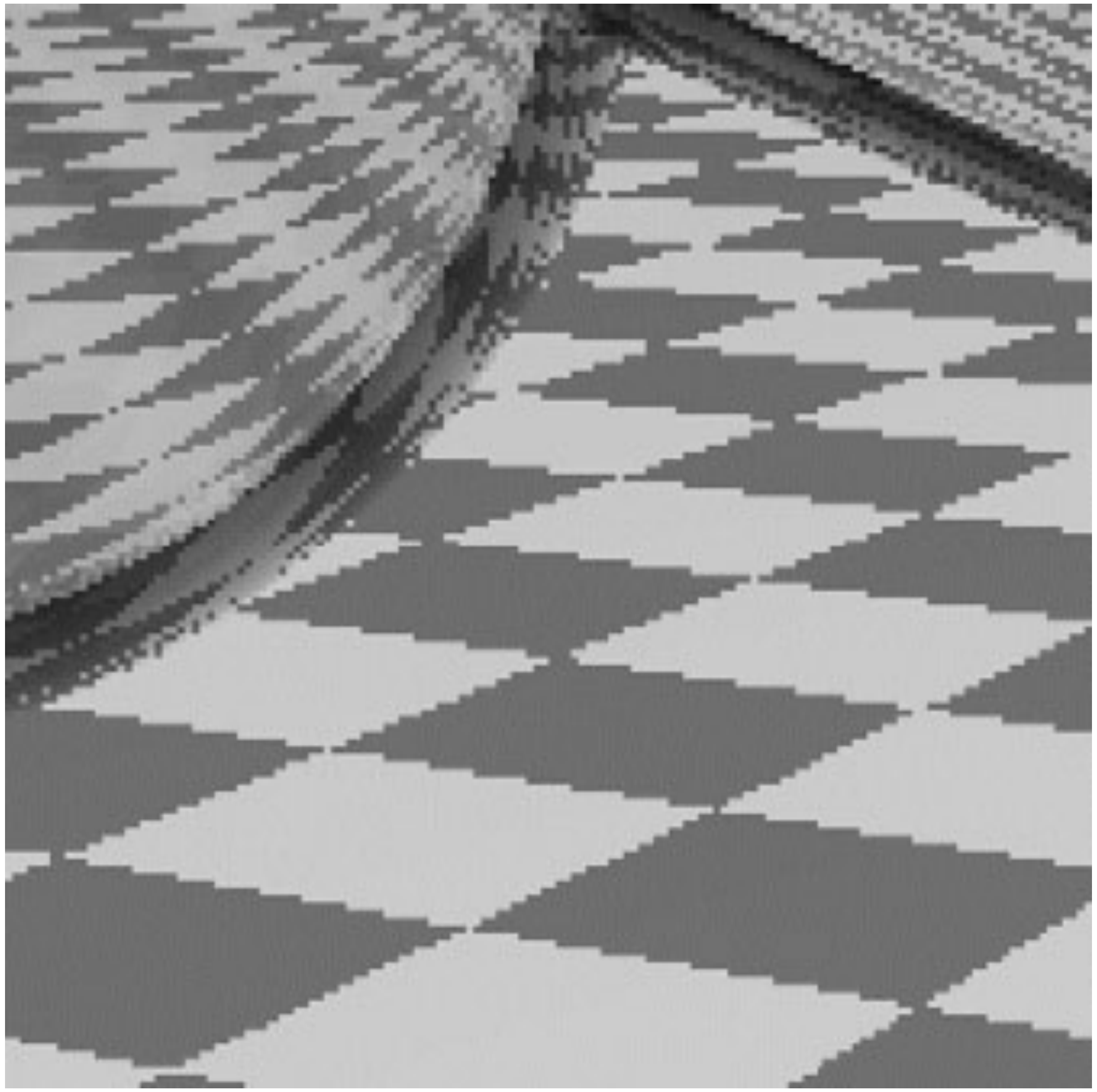




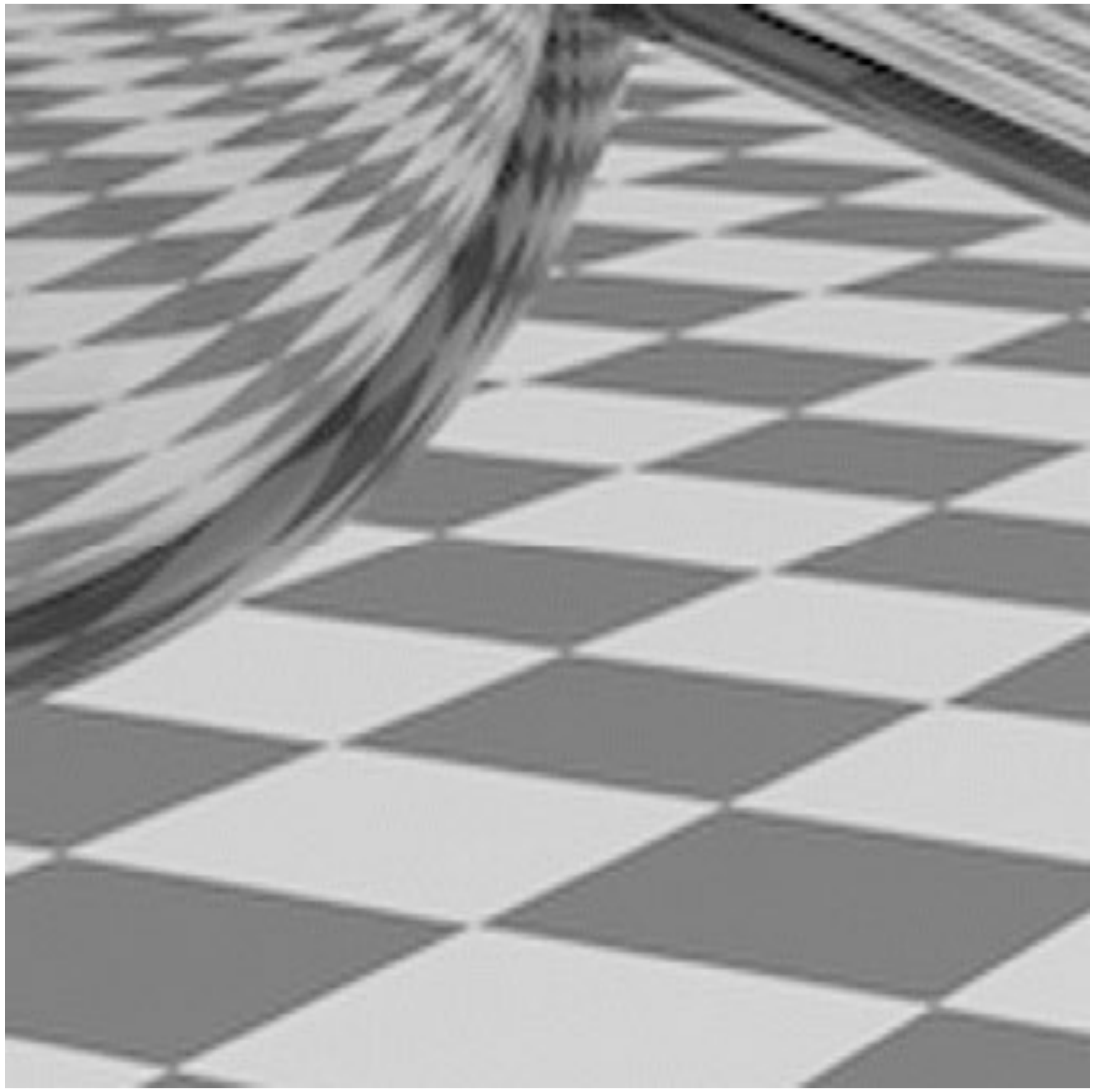
Antialiasing



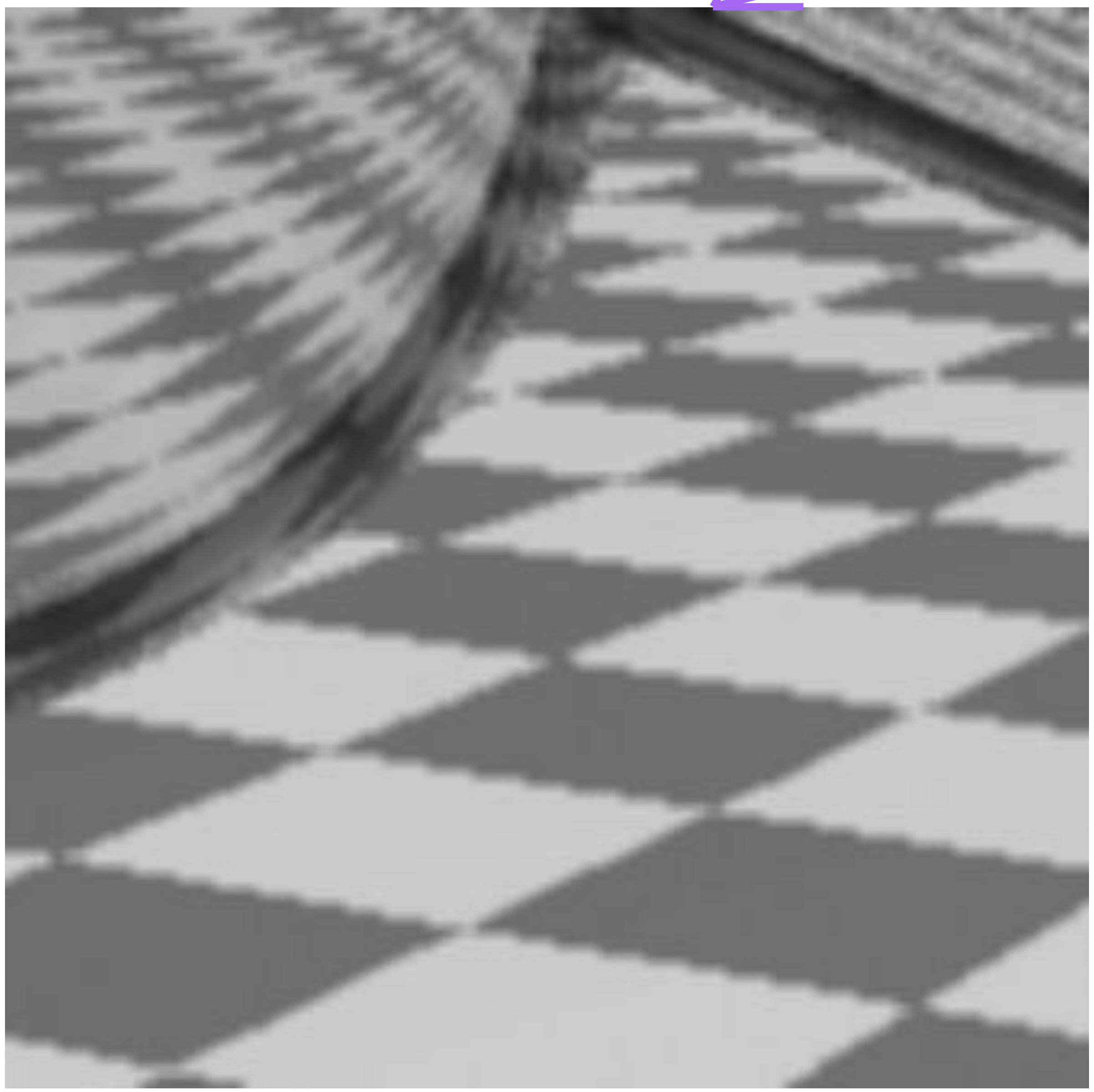
Point Sampling vs Antialiasing



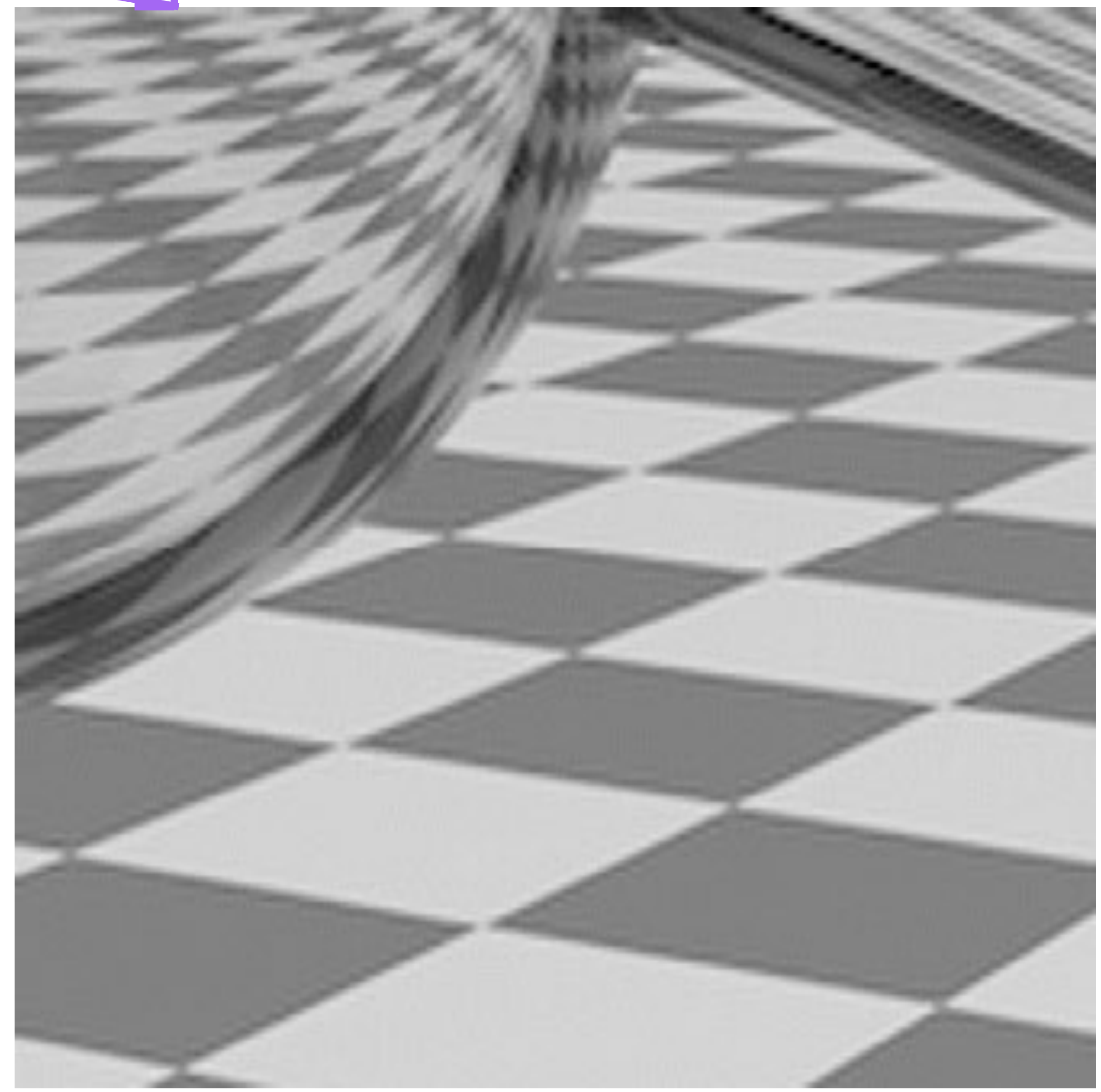
滤波后采样



Antialiasing vs Blurred Aliasing



(Sample then filter, WRONG!)



(Filter then sample)

But why?

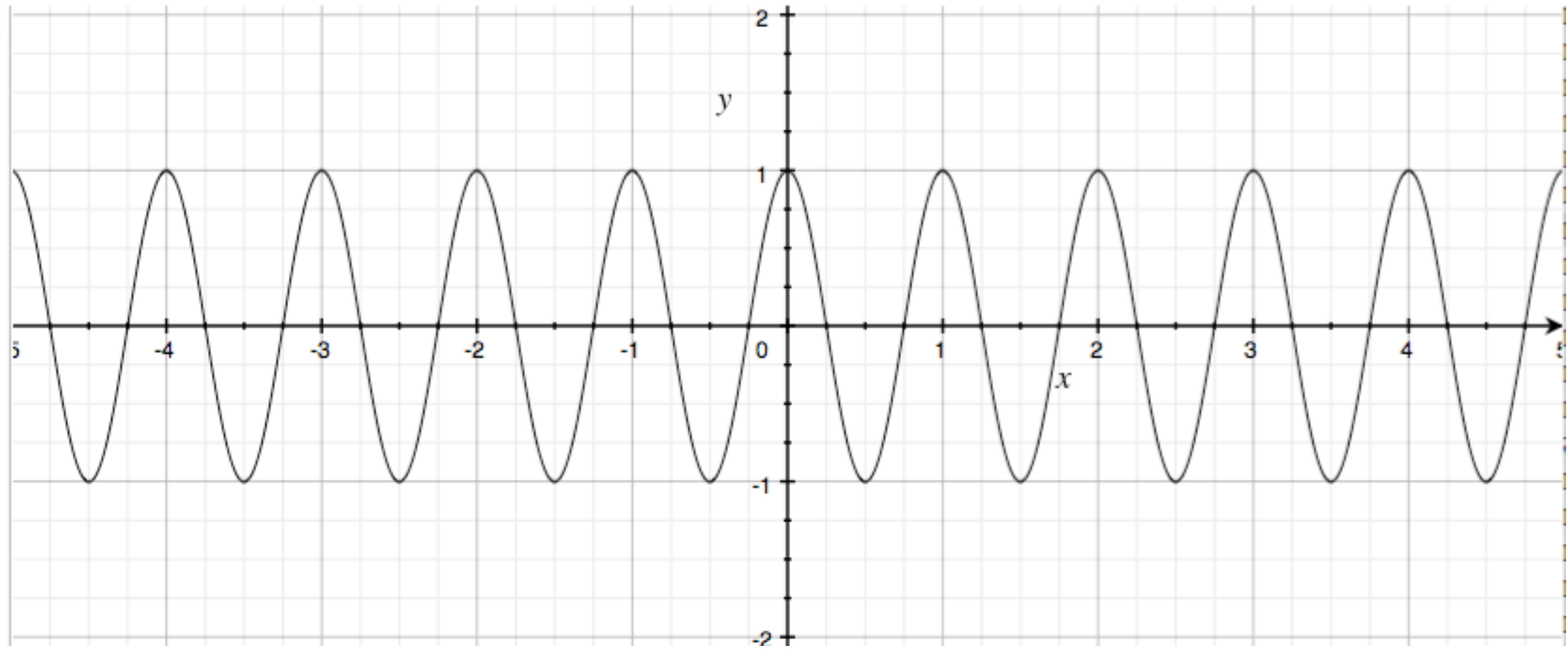
1. Why undersampling introduces aliasing?
2. Why pre-filtering then sampling can do antialiasing?

Let's dig into fundamental reasons

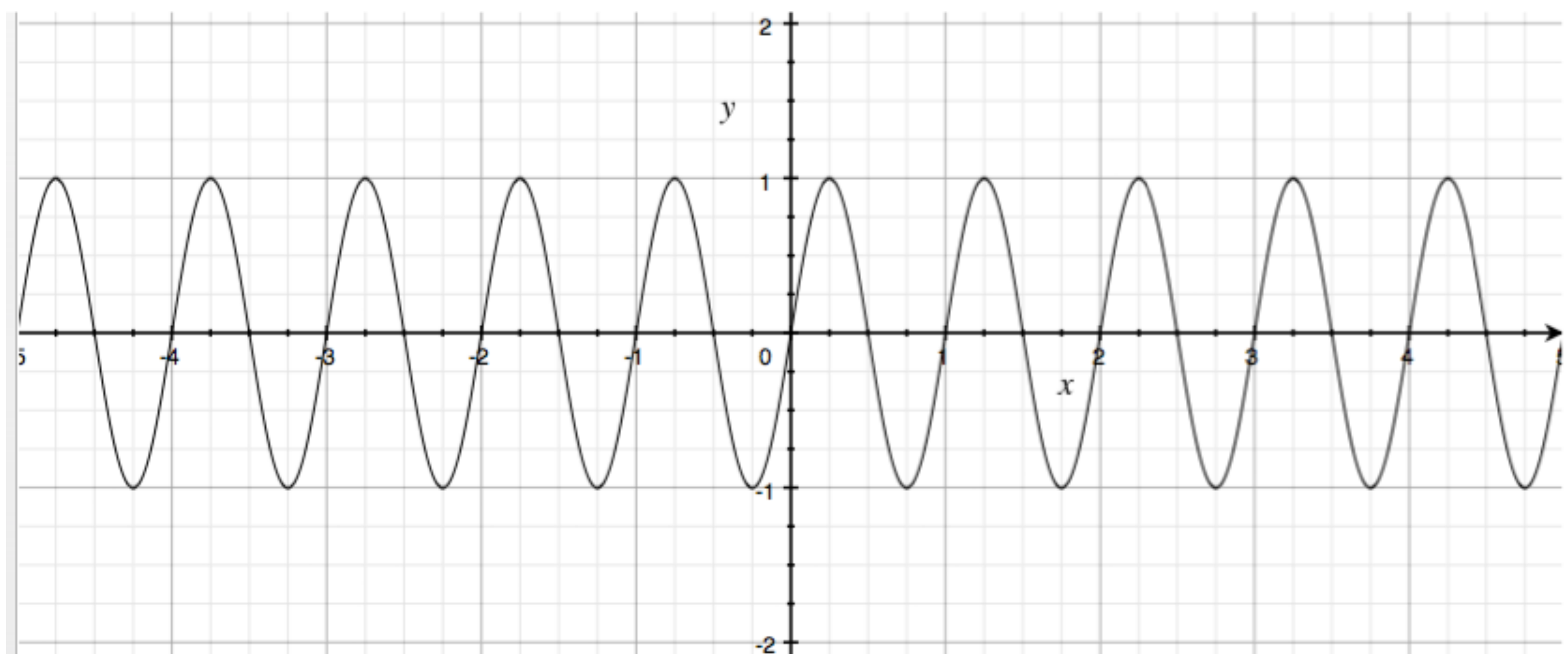
And look at how to implement antialiased rasterization

Frequency Domain

Sines and Cosines



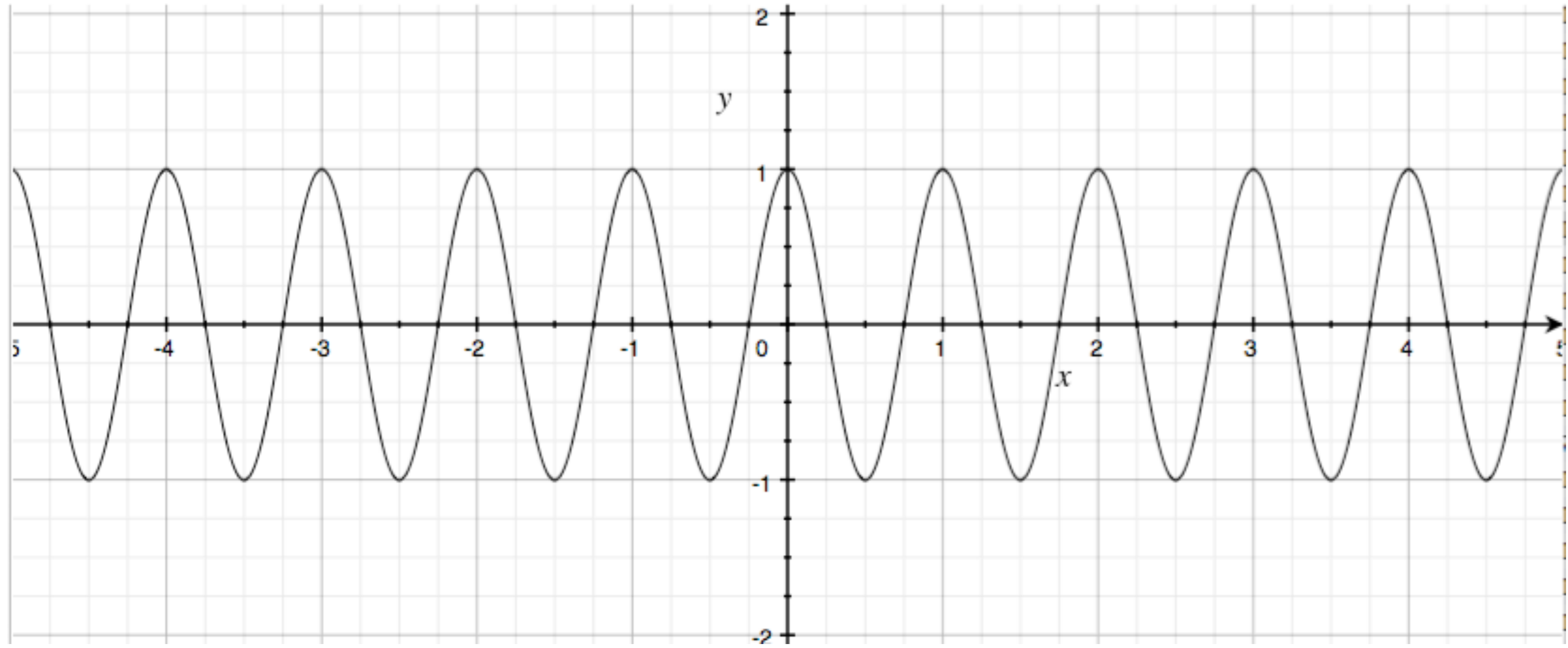
$$\cos 2\pi x$$



$$\sin 2\pi x$$

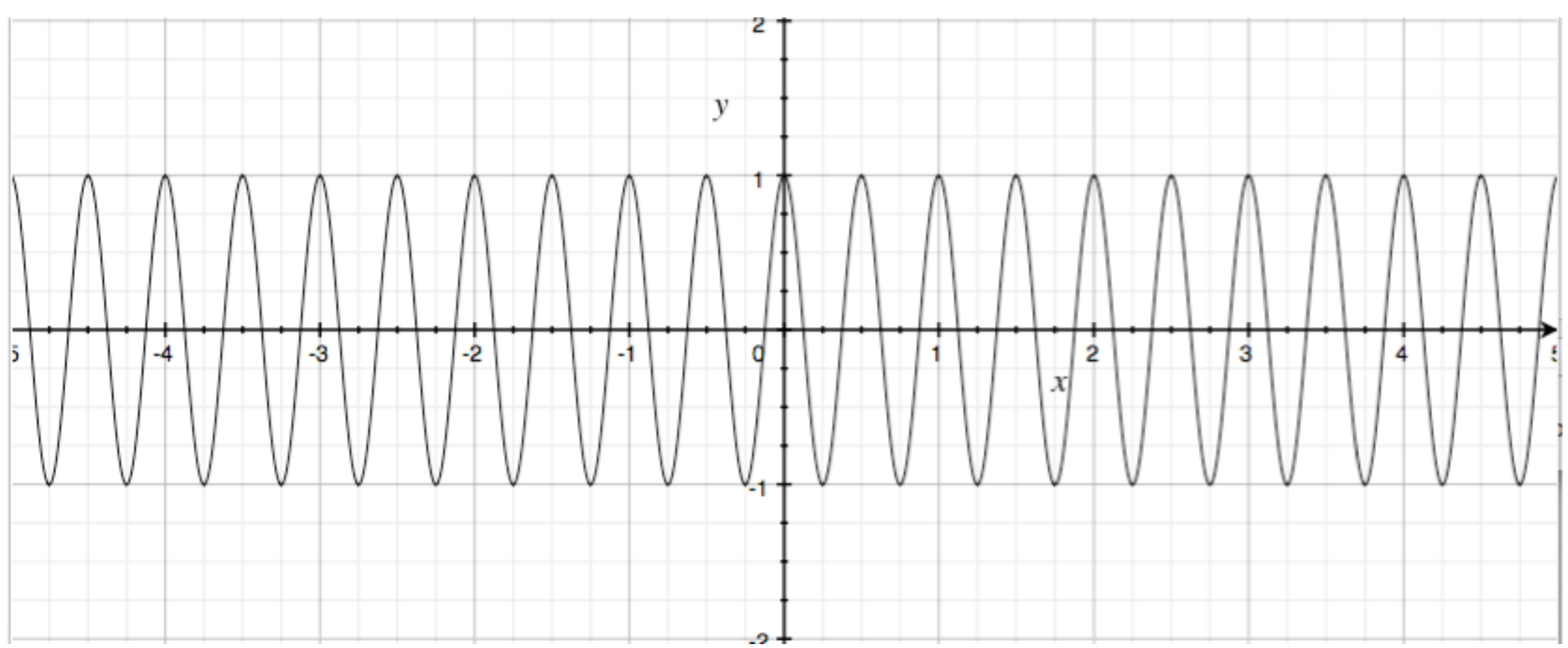
Frequencies $\cos 2\pi f x$

$$f = \frac{1}{T}$$



$f = 1$

$\cos 2\pi x$



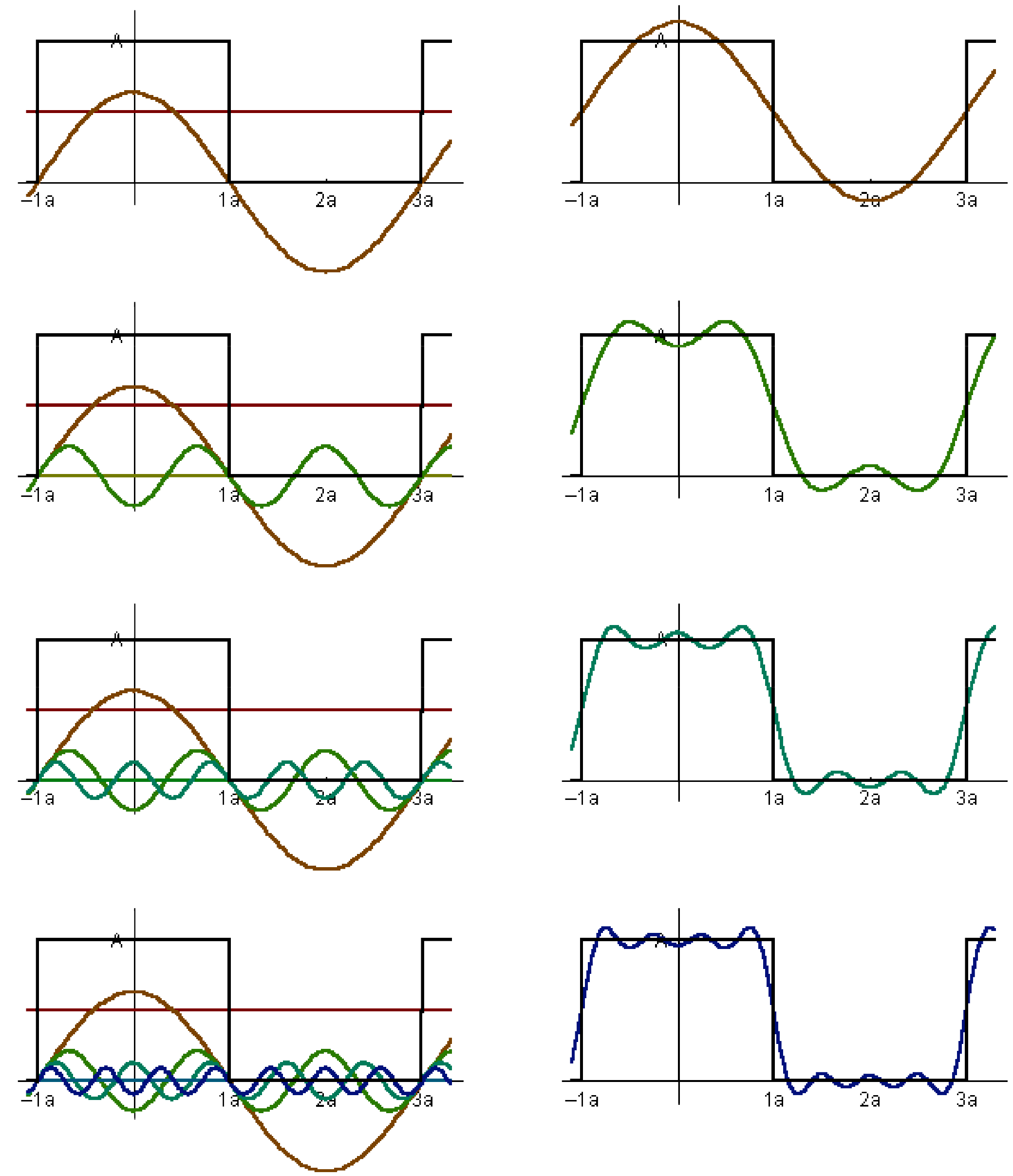
$f = 2$

$\cos 4\pi x$

Fourier Transform

Represent a function as a weighted sum of sines and cosines

Joseph Fourier 1768 - 1830



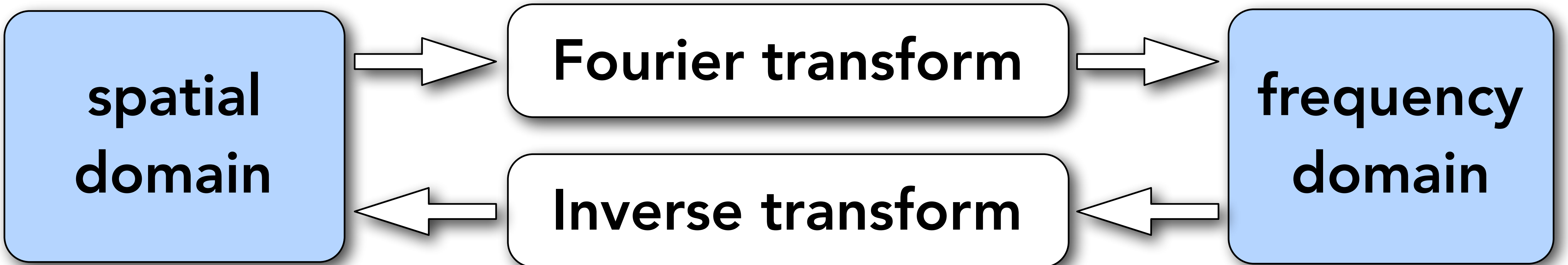
将周期函数展开成由三角函数组成的级数

$$f(x) = \frac{A}{2} + \frac{2A\cos(t\omega)}{\pi} - \frac{2A\cos(3t\omega)}{3\pi} + \frac{2A\cos(5t\omega)}{5\pi} - \frac{2A\cos(7t\omega)}{7\pi} + \cdots$$

任何一个函数可以分解成不同的频率

Fourier Transform Decomposes A Signal Into Frequencies

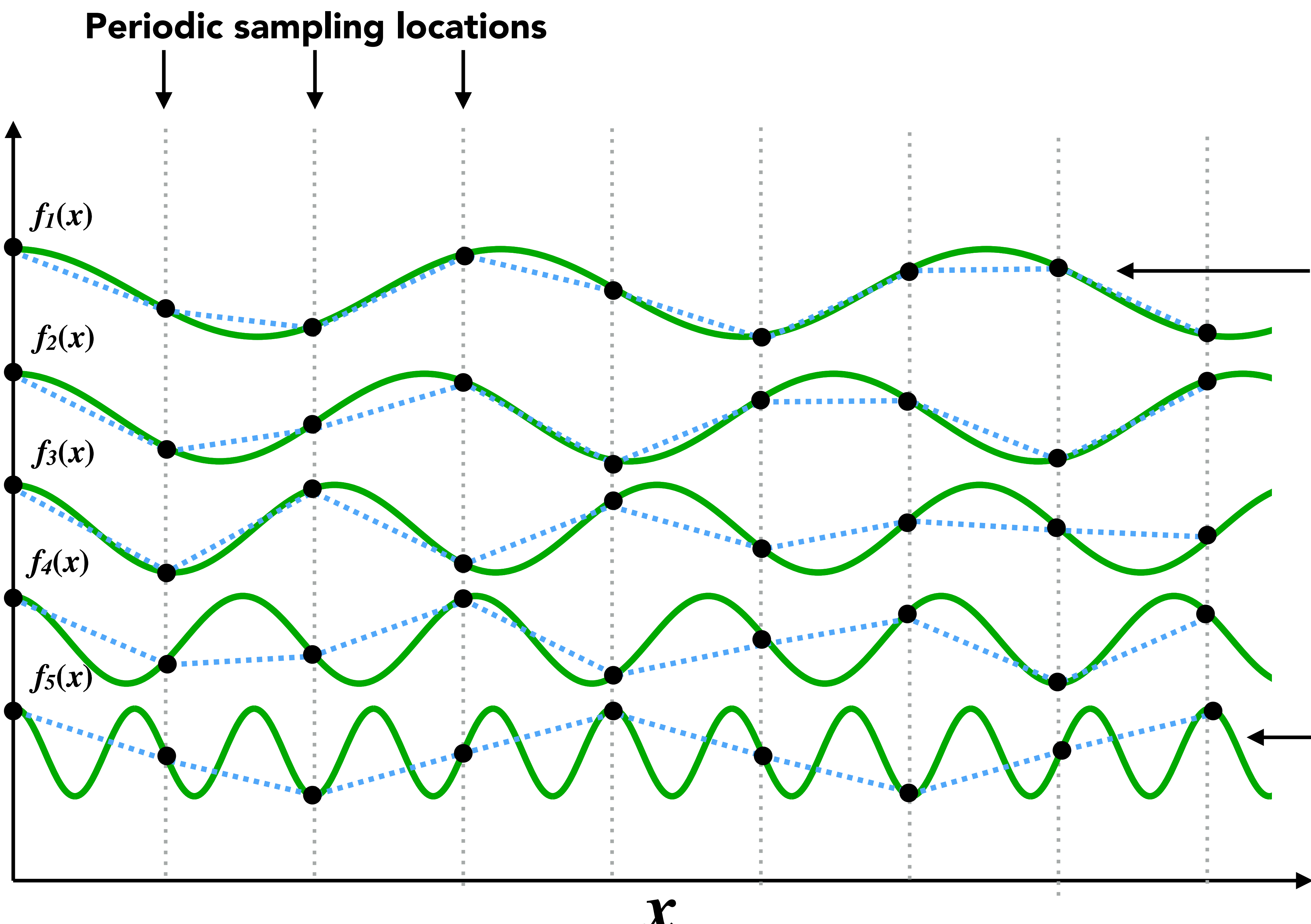
$$f(x) \qquad F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i \omega x} dx \qquad F(\omega)$$



$$f(x) = \int_{-\infty}^{\infty} F(\omega) e^{2\pi i \omega x} d\omega$$

Recall $e^{ix} = \cos x + i \sin x$

Higher Frequencies Need Faster Sampling

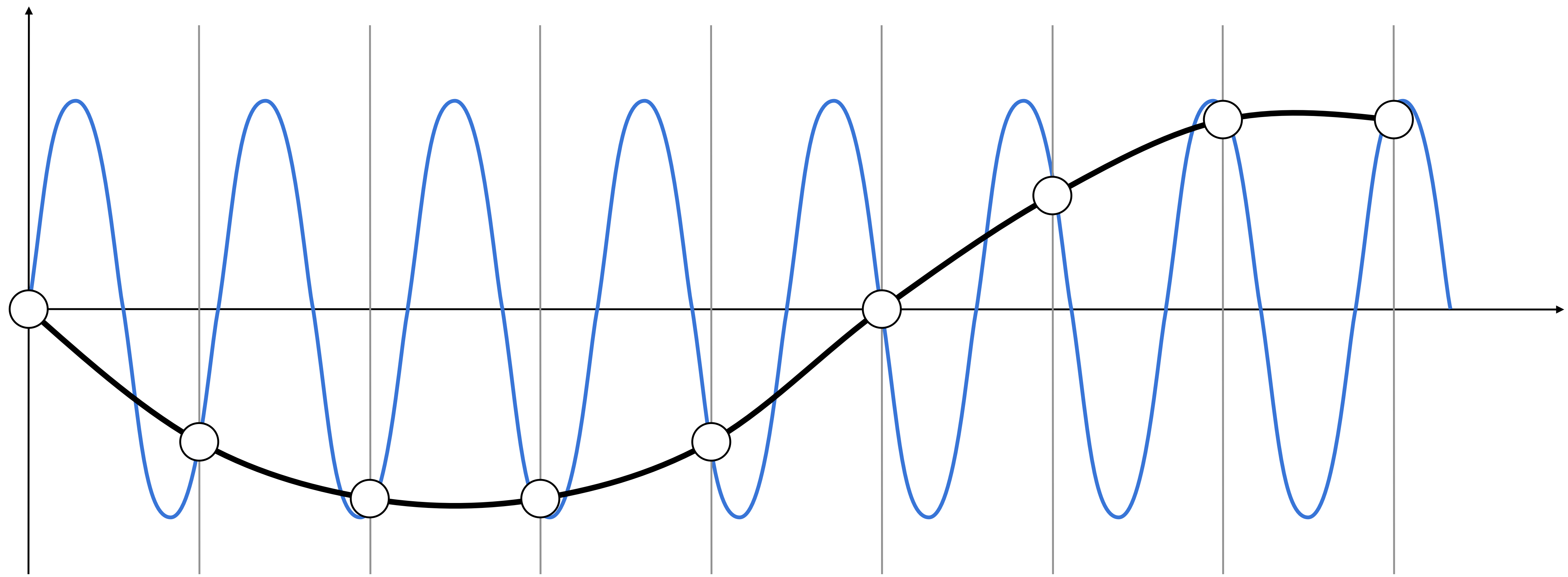


Low-frequency signal: sampled adequately for reasonable reconstruction

High-frequency signal is insufficiently sampled: reconstruction incorrectly appears to be from a low frequency signal

函数频率高，采样频率低，就很难还原回函数原本的样子

Undersampling Creates Frequency Aliases



High-frequency signal is insufficiently sampled: samples erroneously appear to be from a low-frequency signal

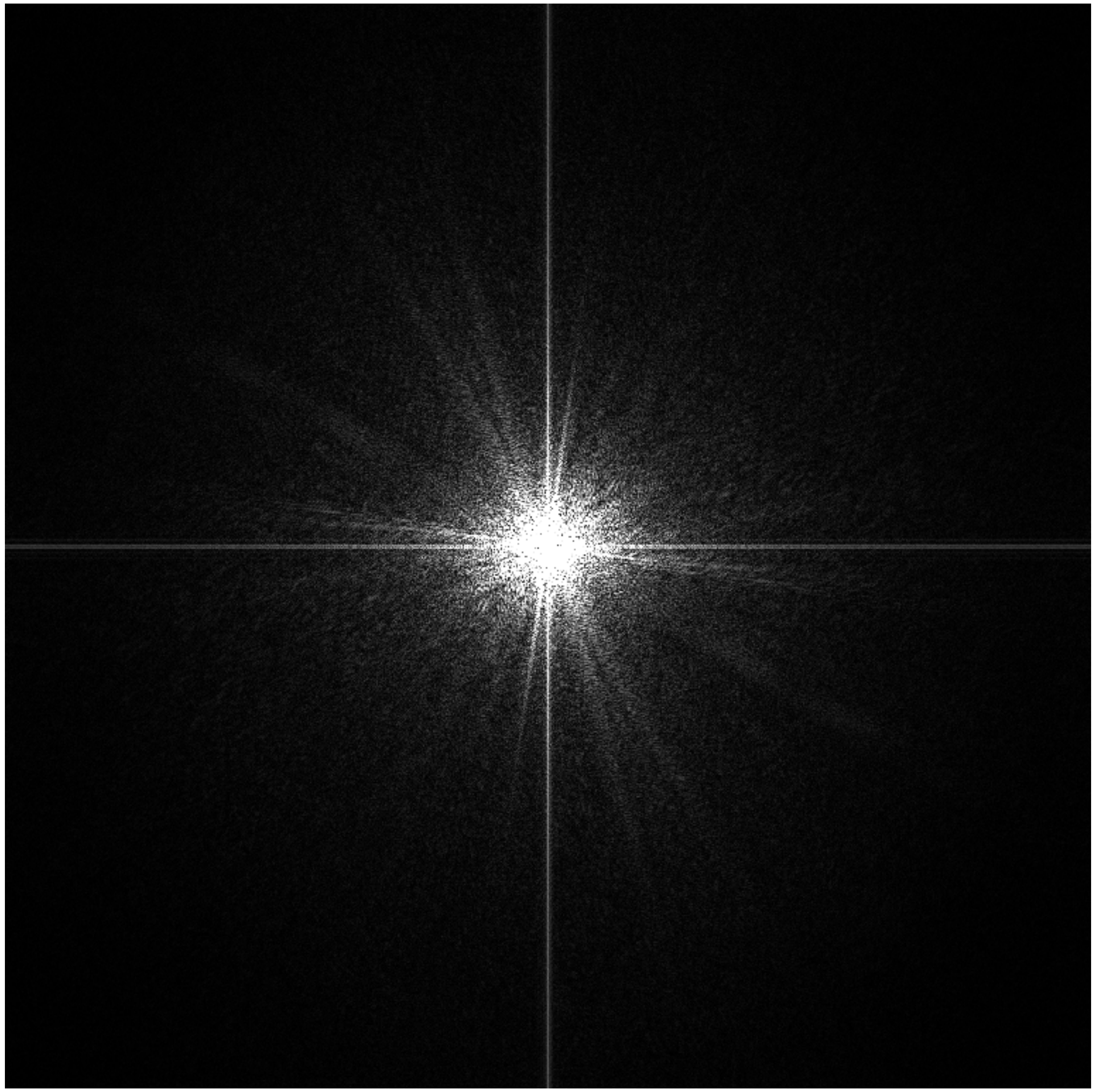
Two frequencies that are indistinguishable at a given sampling rate are called “aliases”

使用同一种采样方法，采样两个完全不同的函数，得到的结果
完全一样，被称为走样

Filtering = Getting rid of certain frequency contents

去掉特定的频率

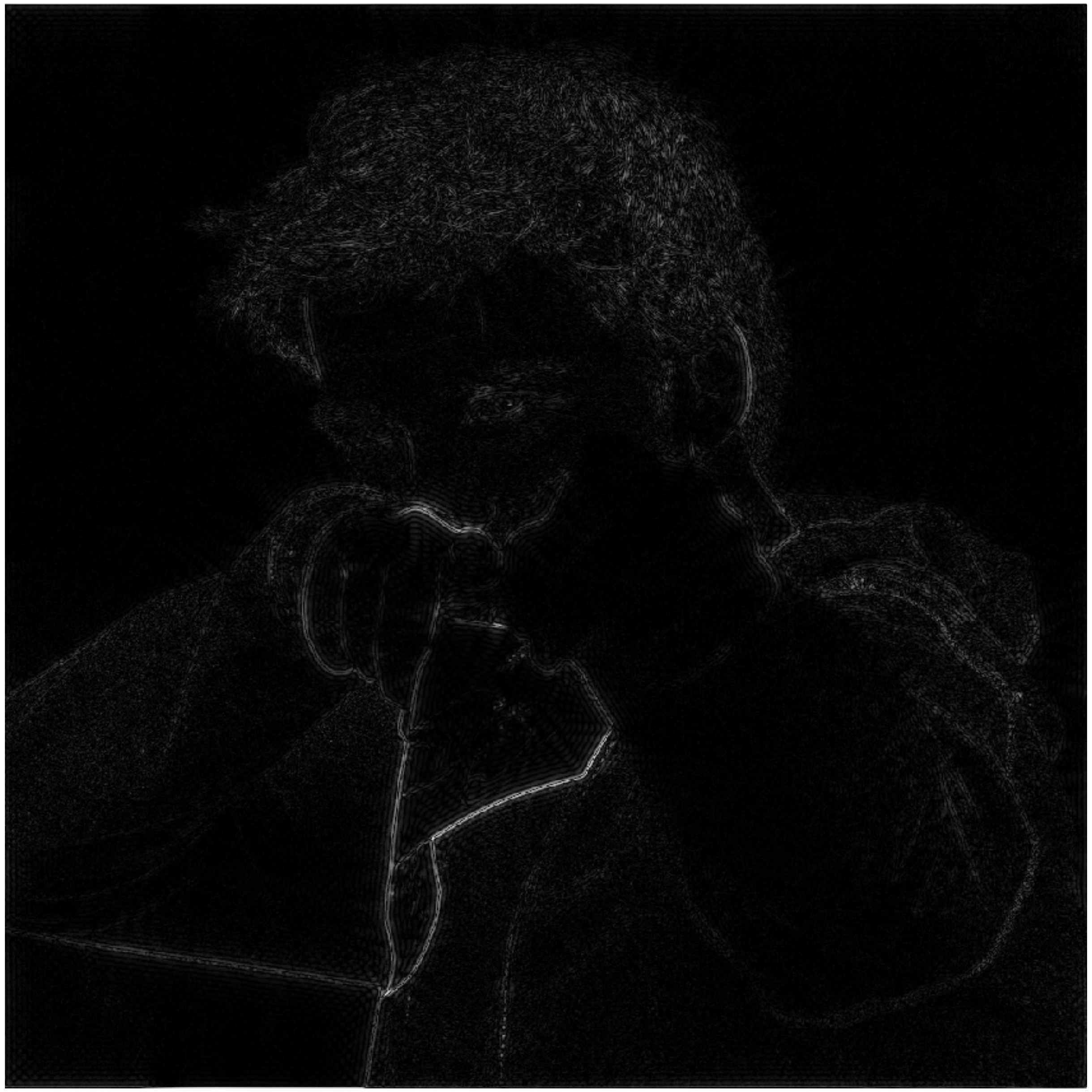
Visualizing Image Frequency Content

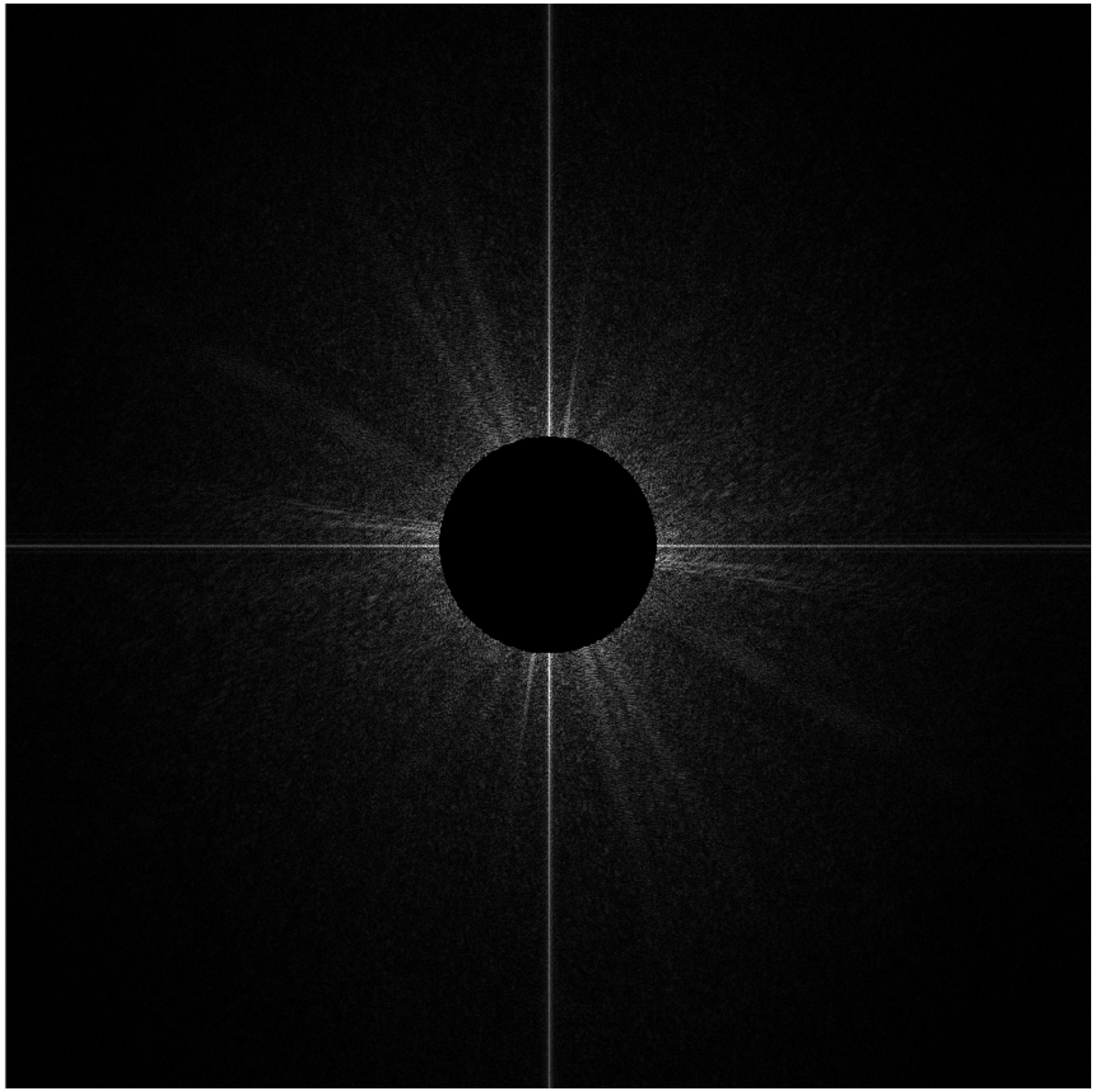


从时域变换到频域

假设无限多的照片边挨着边平铺，在两个照片相交的
边缘位置会发生剧烈的信号变化

Filter Out Low Frequencies Only (Edges)

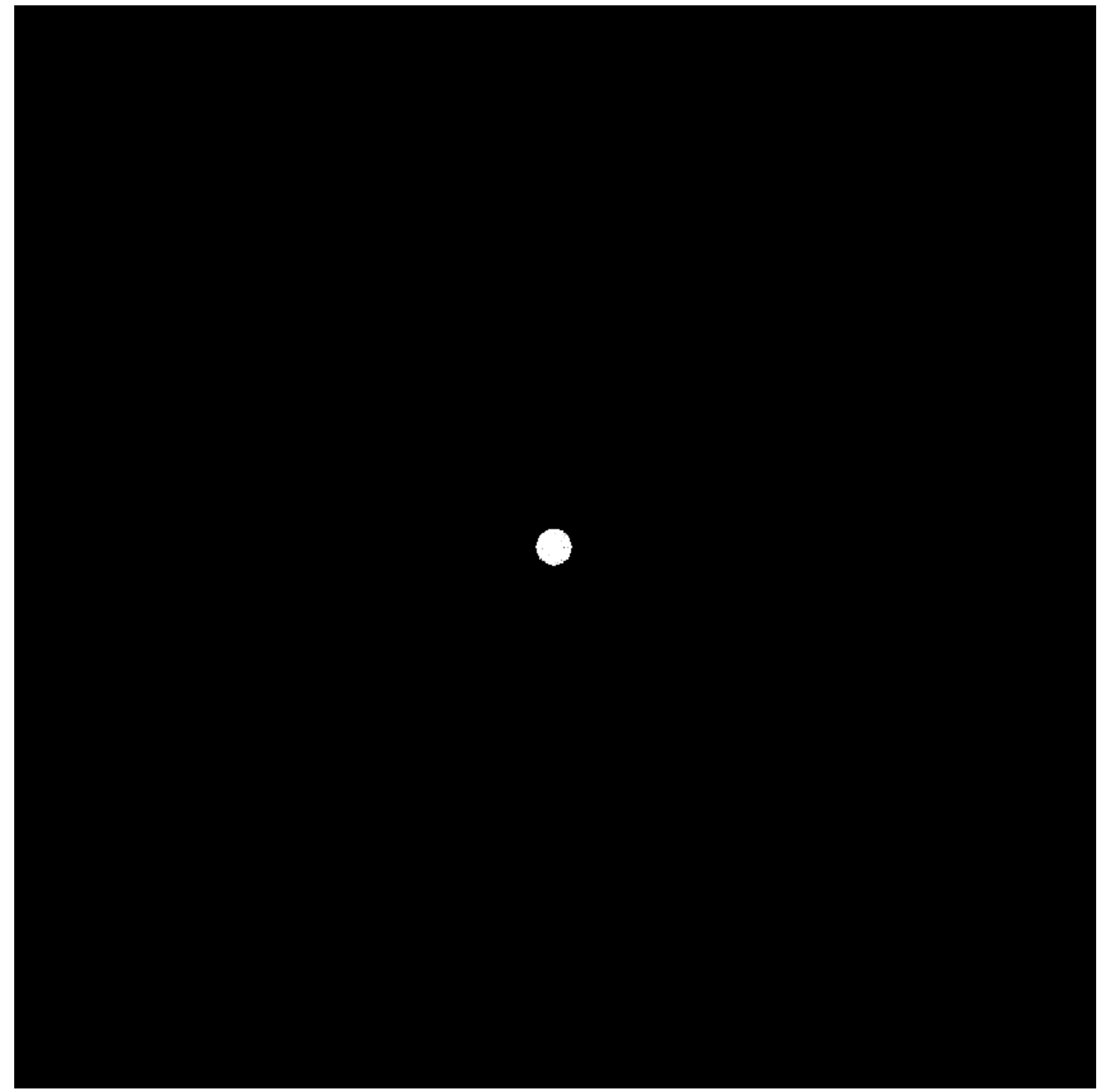




High-pass filter

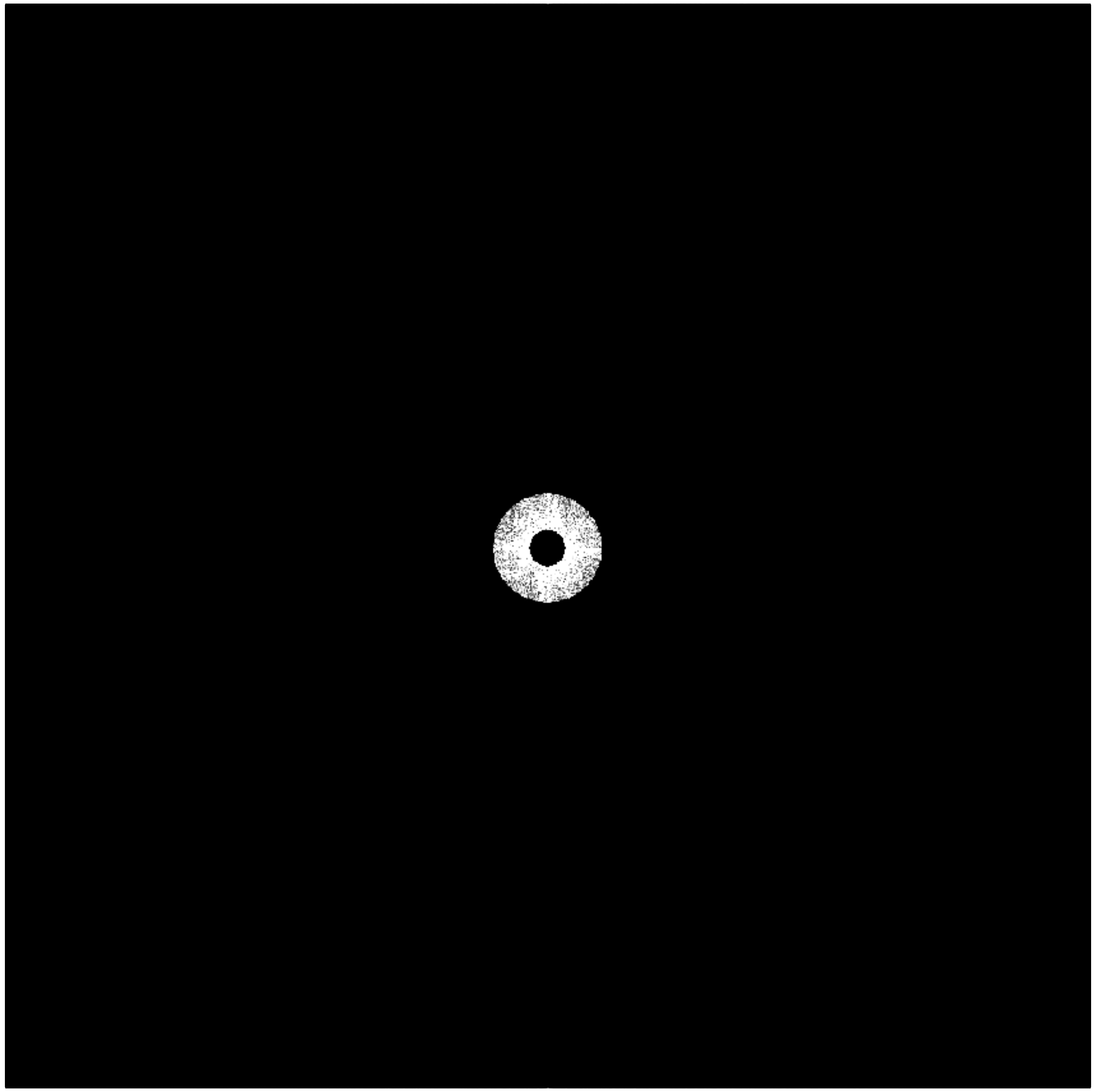
边缘交界处信号变化大

Filter Out High Frequencies (Blur)

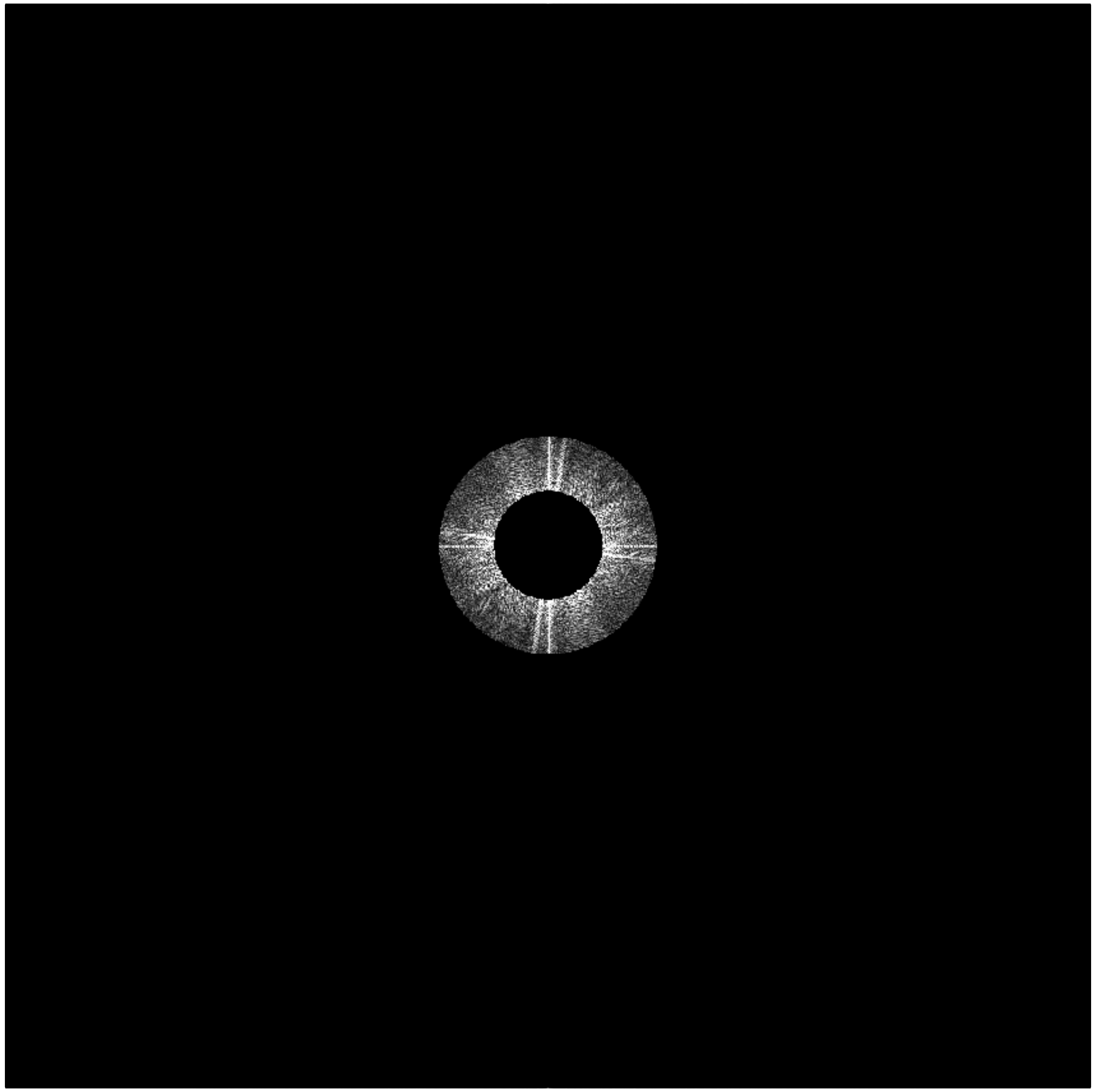


Low-pass filter

Filter Out Low and High Frequencies



Filter Out Low and High Frequencies



数字图像处理--相关课程

Filtering = Convolution 卷积

(= Averaging)

Convolution

Signal

1	3	5	3	7	1	3	8	6	4

Filter

1/4	1/2	1/4

Point-wise local averaging in a “sliding window”

Convolution

Signal

1	3	5	3	7	1	3	8	6	4

Filter

1/4	1/2	1/4

$1 \times (1/4) + 3 \times (1/2) + 5 \times (1/4) = 3$

Result

	3								

Convolution

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							

对周围加权平均的结果

Convolution Theorem

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

时域上对两个信号进行卷积反映在频域上是两个信号的频域的乘积

Option 1:

如何做一个卷积?

·可以拿到一幅图用一个卷积的滤波器做一个卷积操作

·也可以将图傅里叶变换到频域上，再将卷积的滤波器变换到频域上，将两者进行相乘得到频域的结果再逆傅里叶变换到时域上

- 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)

在时域上的乘积相当于频域上的卷积

Convolution Theorem

Spatial Domain

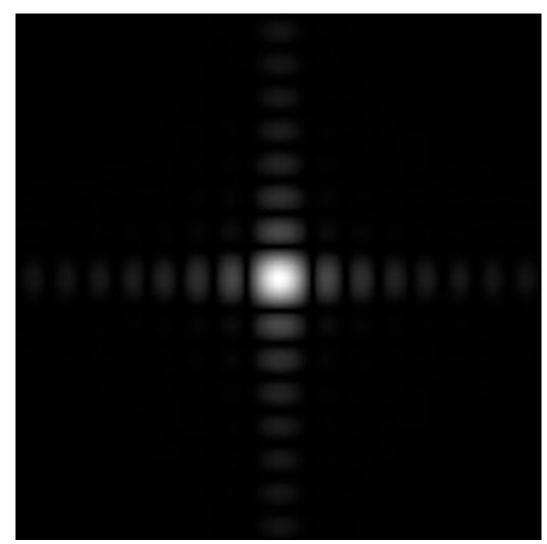
$$* \ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix} =$$

Fourier Transform ↓ ↓

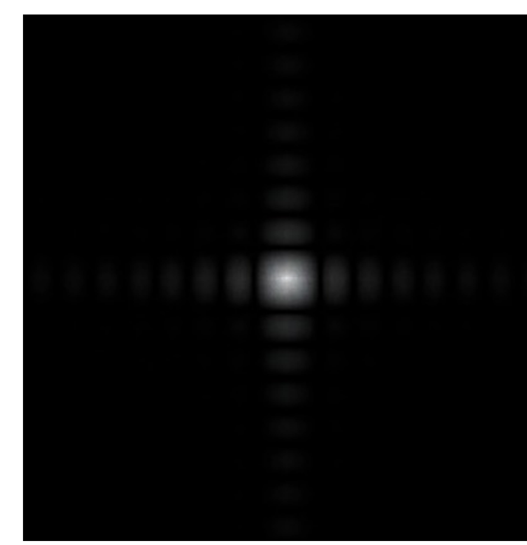
Inv. Fourier Transform ↑

Frequency Domain

x



=



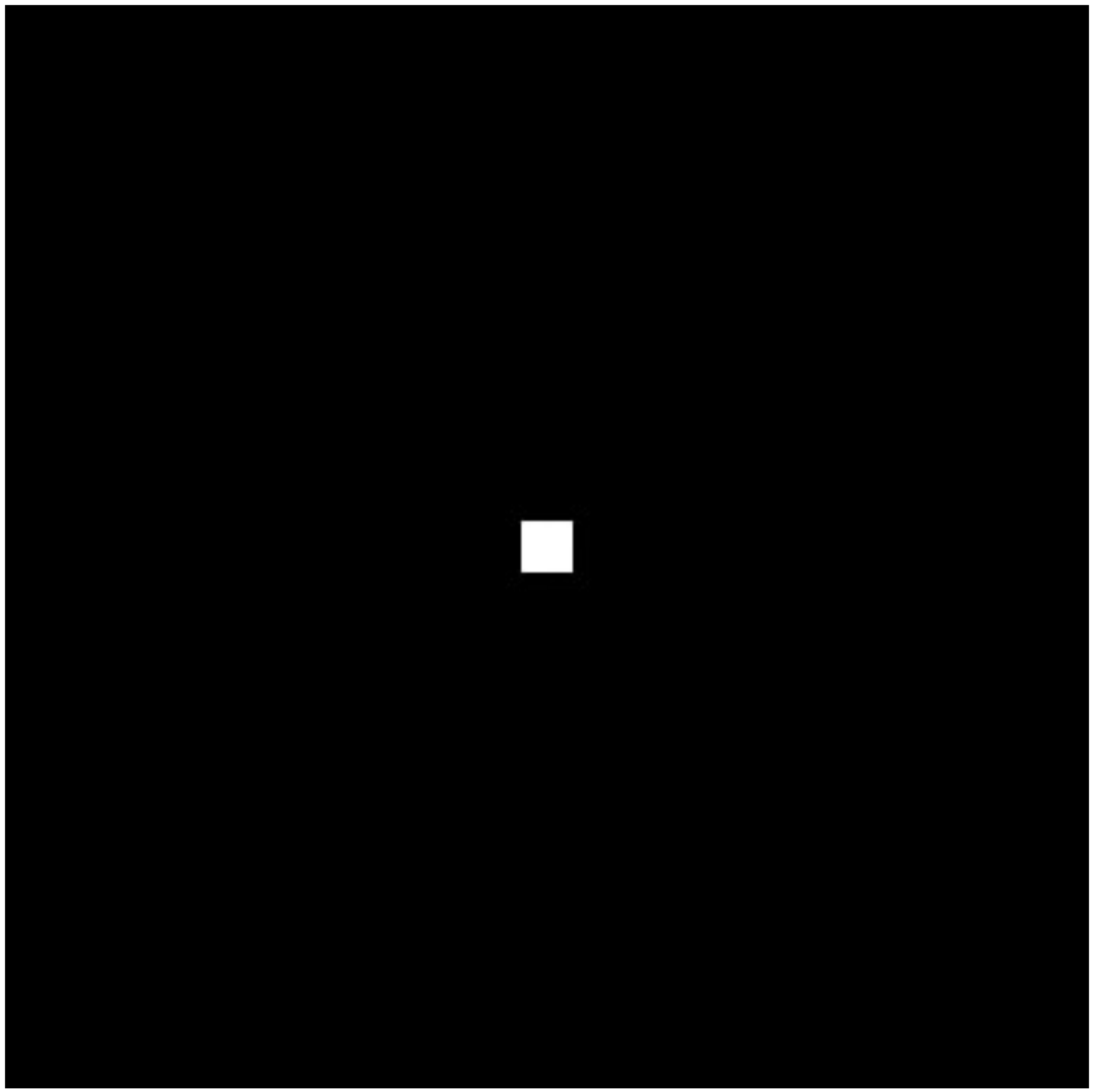
做了一个低通滤波

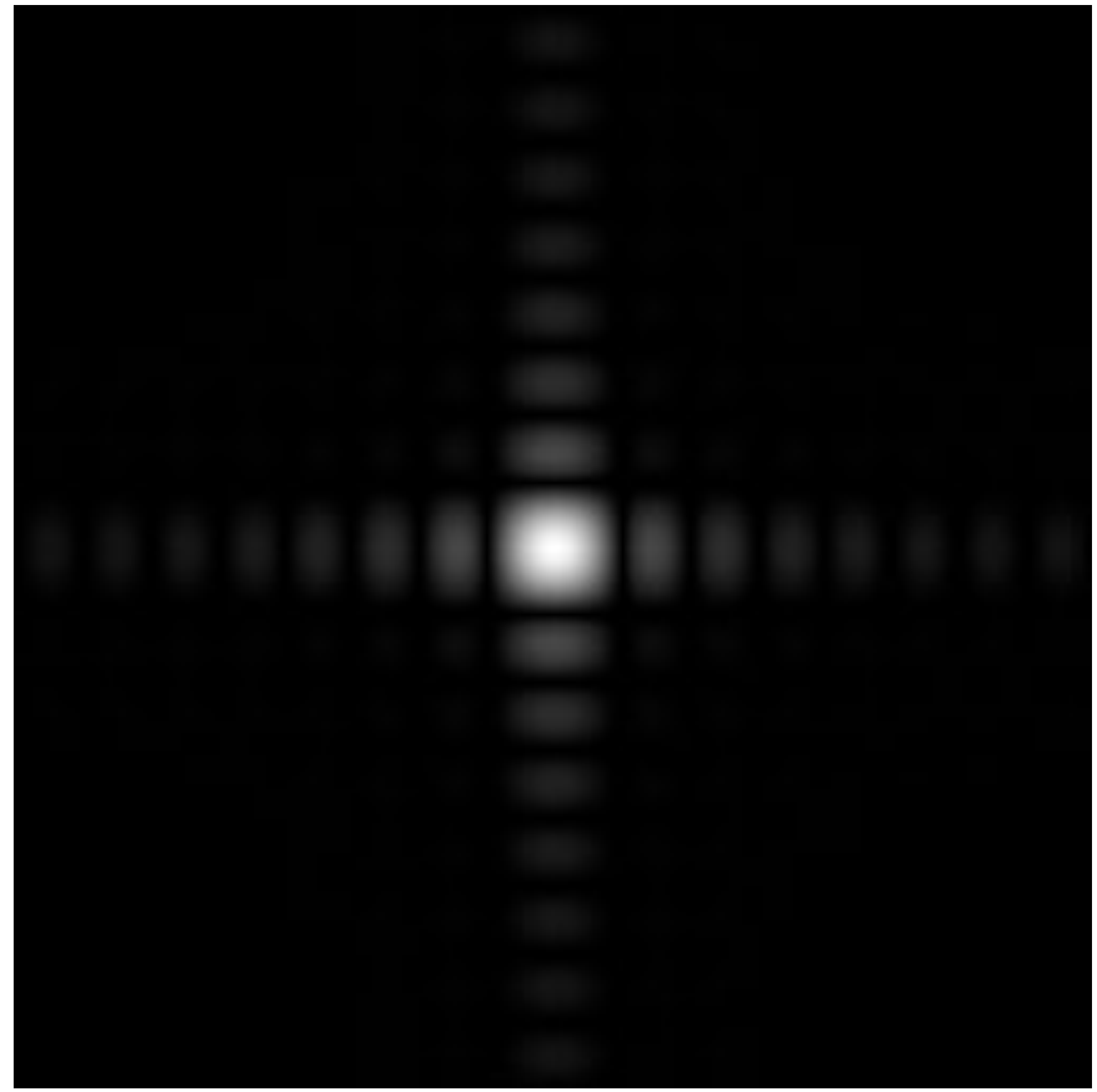
Box Filter

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

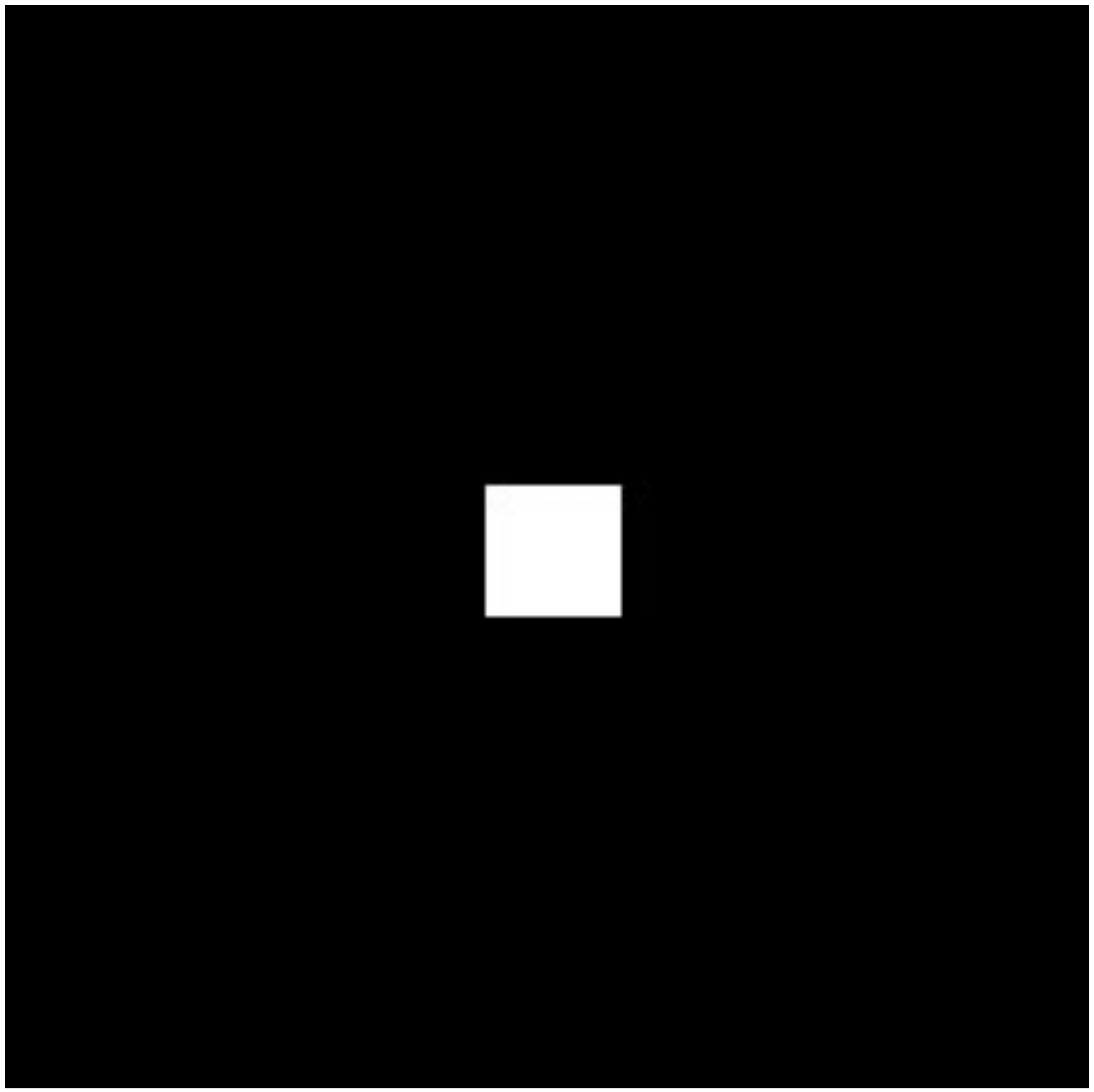
Example: 3x3 box filter

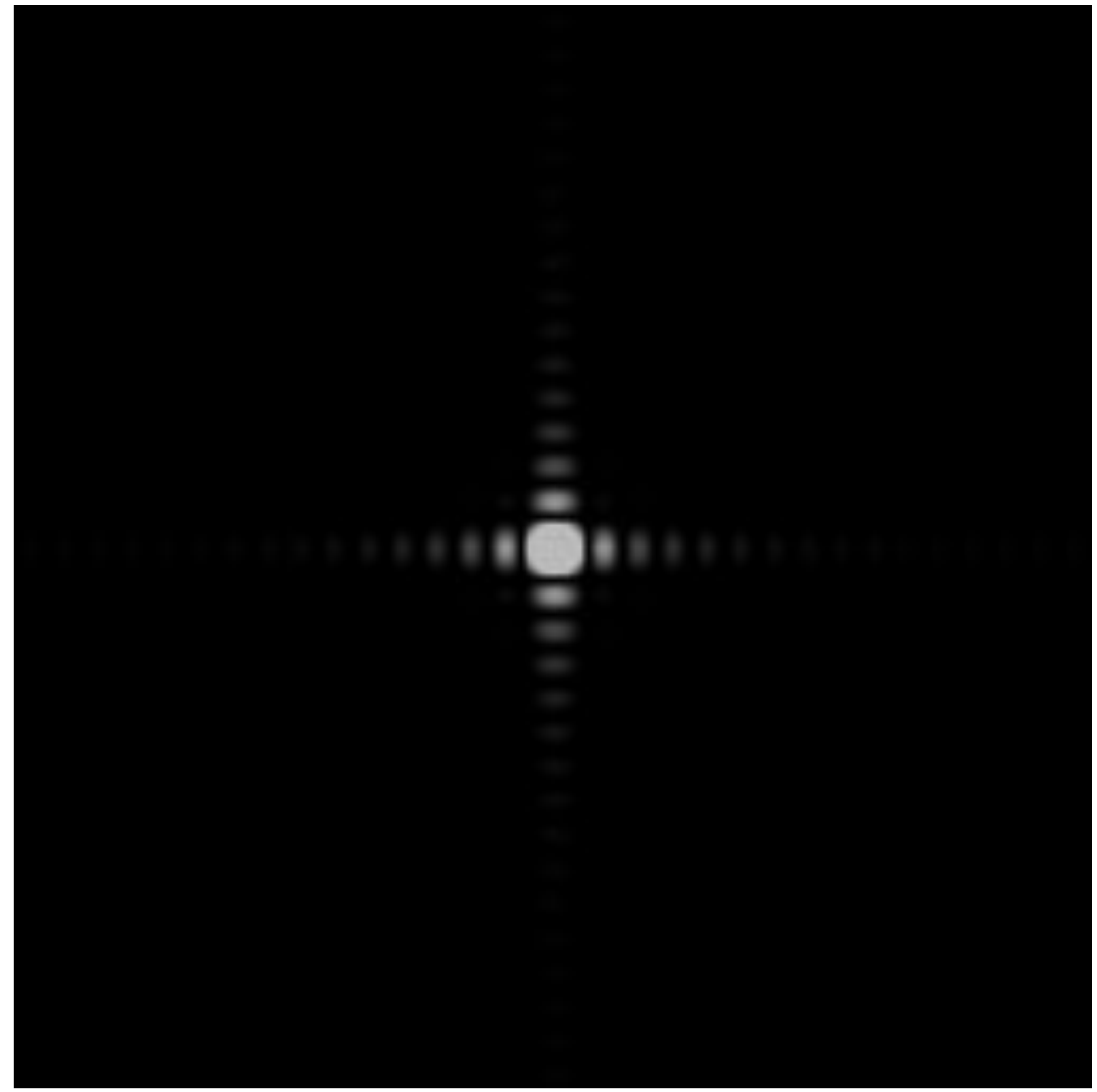
Box Function = "Low Pass" Filter





Wider Filter Kernel = Lower Frequencies



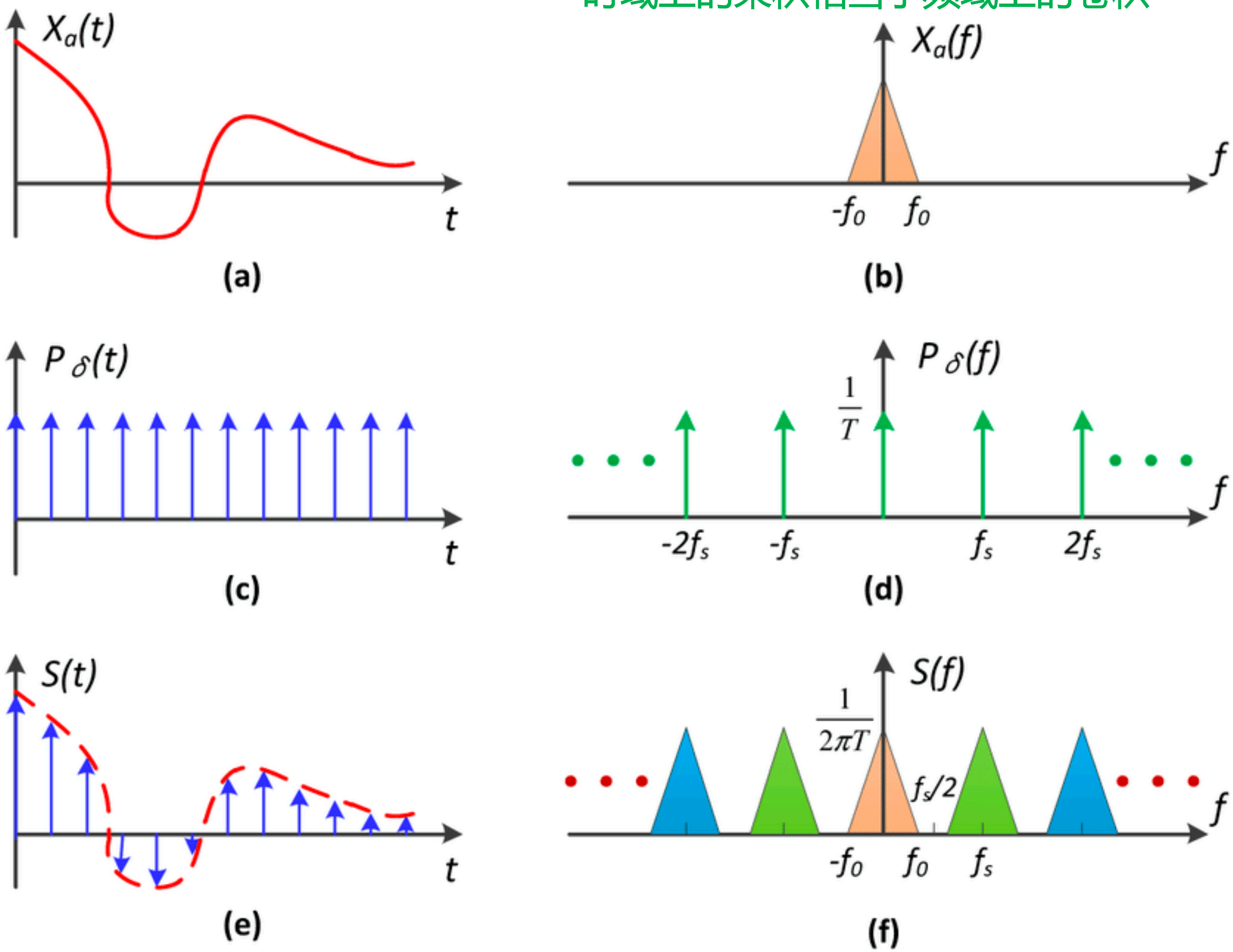


Sampling = Repeating Frequency Contents

Sampling = Repeating Frequency Contents

用函数a乘以函数c得到函数e

时域上的乘积相当于频域上的卷积

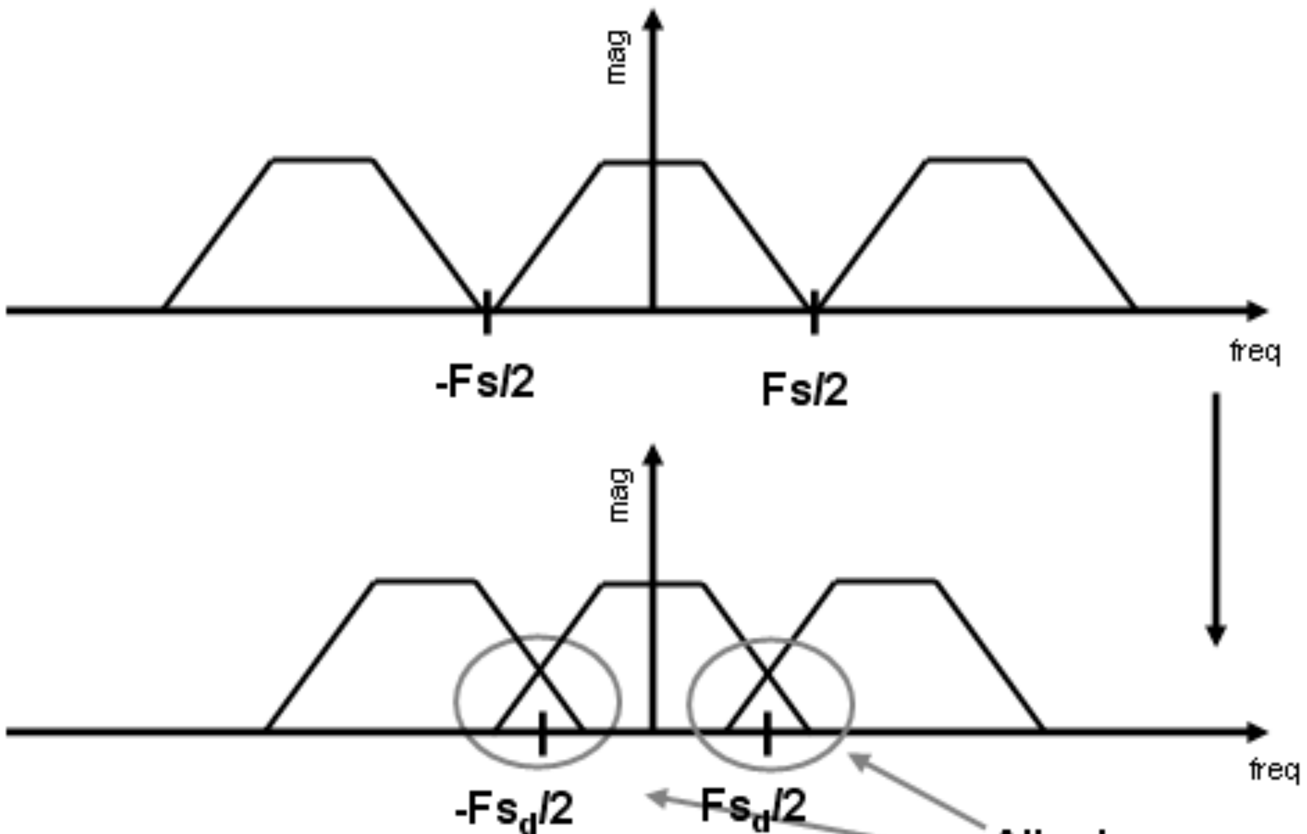


https://www.researchgate.net/figure/The-evolution-of-sampling-theorem-a-The-time-domain-of-the-band-limited-signal-and-b_fig5_301556095

Aliasing = Mixed Frequency Contents

Dense sampling:

Sparse sampling:



采样的越慢，重复的越密集

Antialiasing

How Can We Reduce Aliasing Error?

如像素640*480，像素之间间隔大，现在的视网膜屏幕，像素间隔小，同样一个三角形，采样的频率不同，反映在频域，相当于频谱之间的间隔更大，重叠更小

Option 1: Increase sampling rate

本质上

- Essentially increasing the distance between replicas in the Fourier domain
- Higher resolution displays, sensors, framebuffers…
- But: costly & may need very high resolution

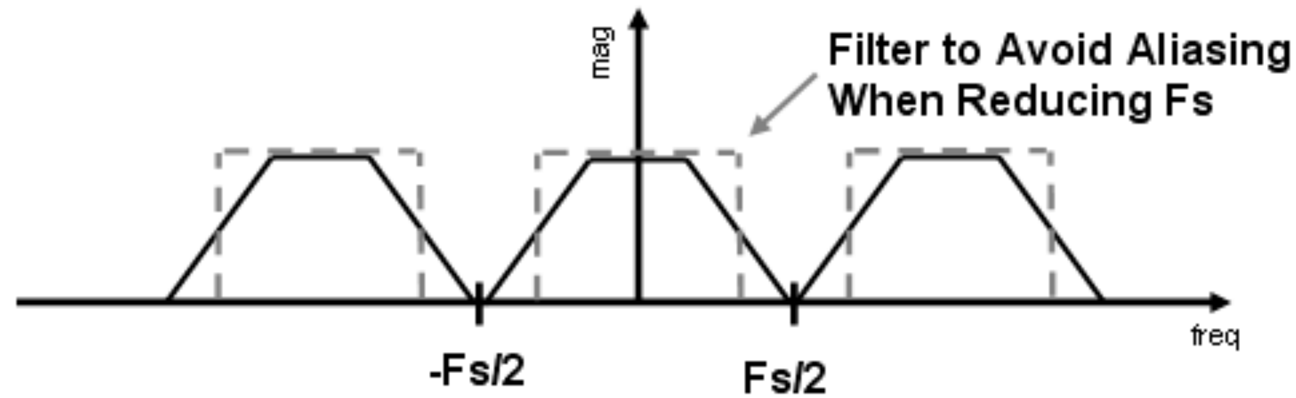
Option 2: Antialiasing

- Making Fourier contents "narrower" before repeating
- i.e. Filtering out high frequencies before sampling

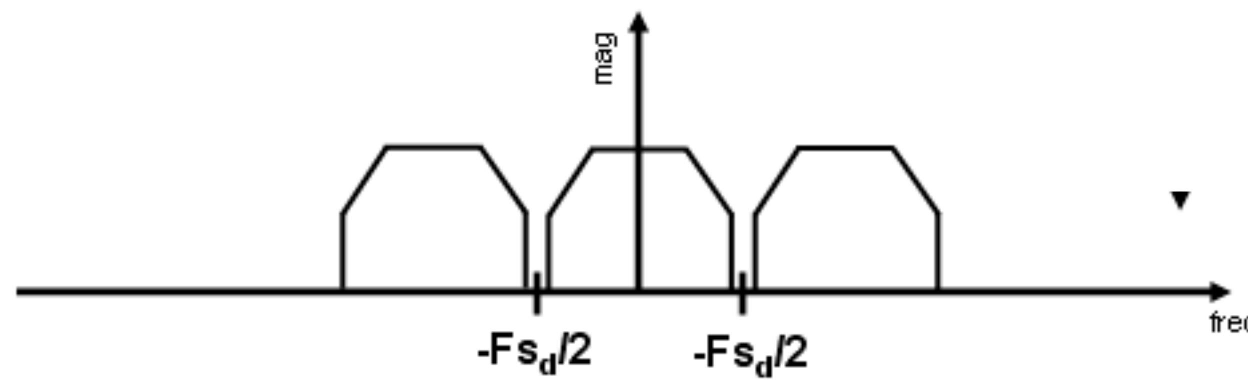
怎么做反走样？先把一个信号的高频信息拿掉再采样

Antialiasing = Limiting, then repeating

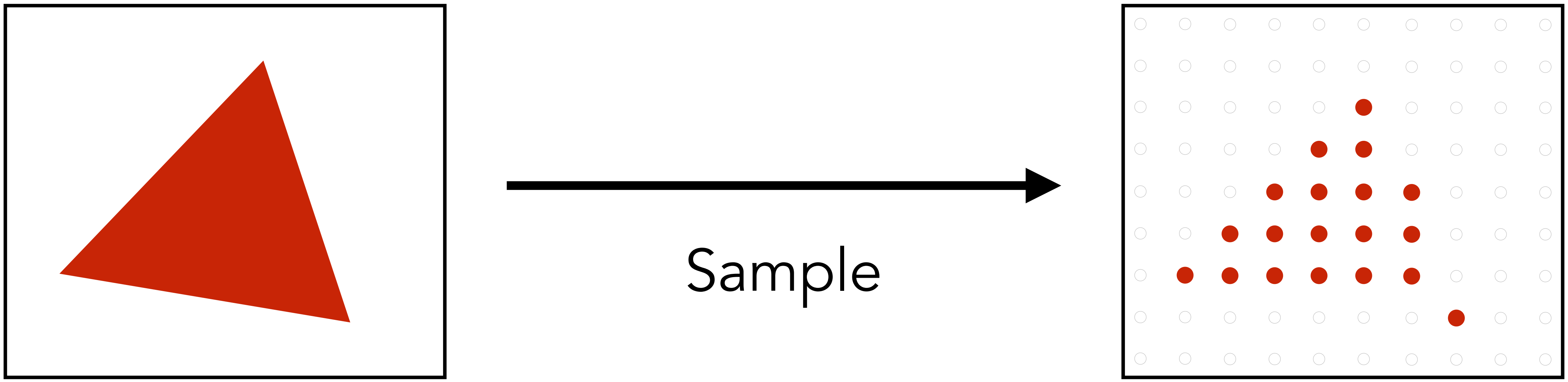
Filtering



Then sparse sampling

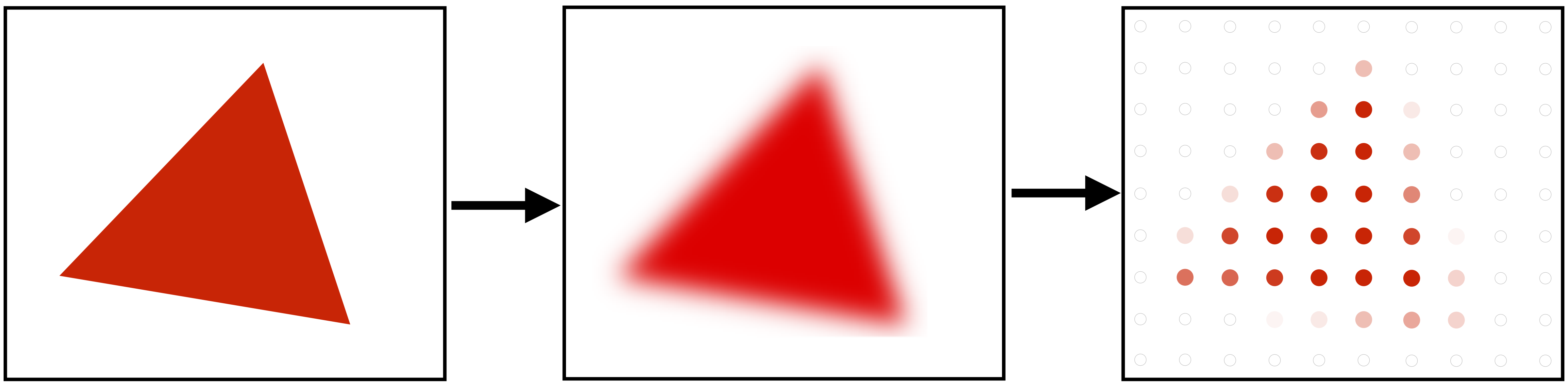


Regular Sampling



Note jaggies in rasterized triangle
where pixel values are pure red or white

Antialiased Sampling



Pre-Filter

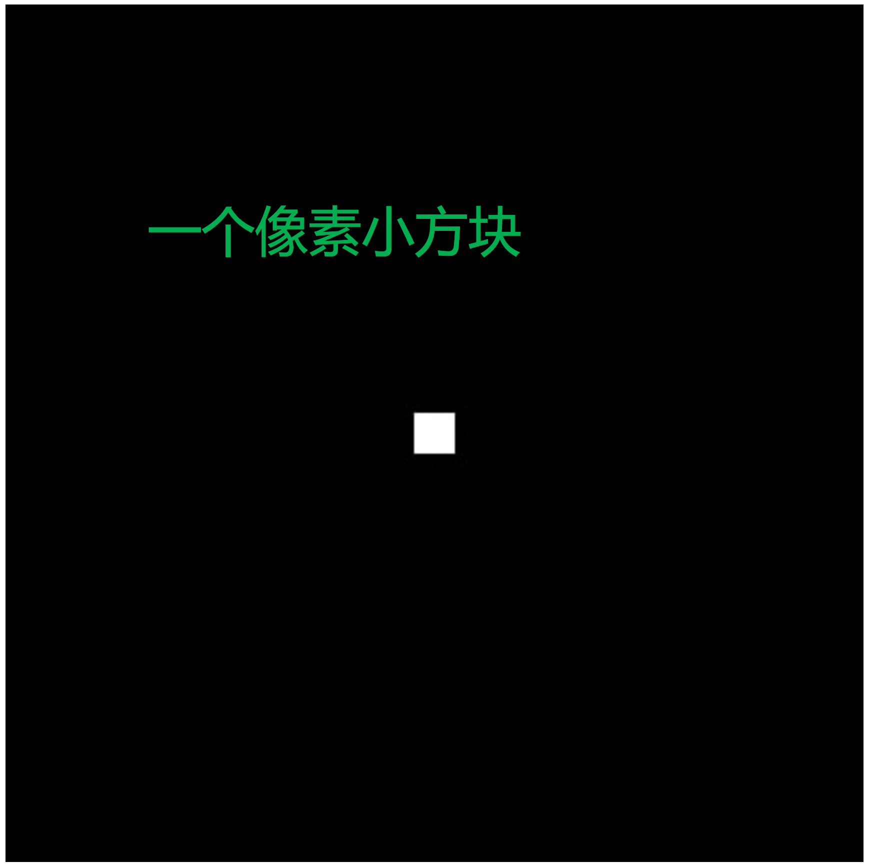
(remove frequencies above Nyquist)

Sample

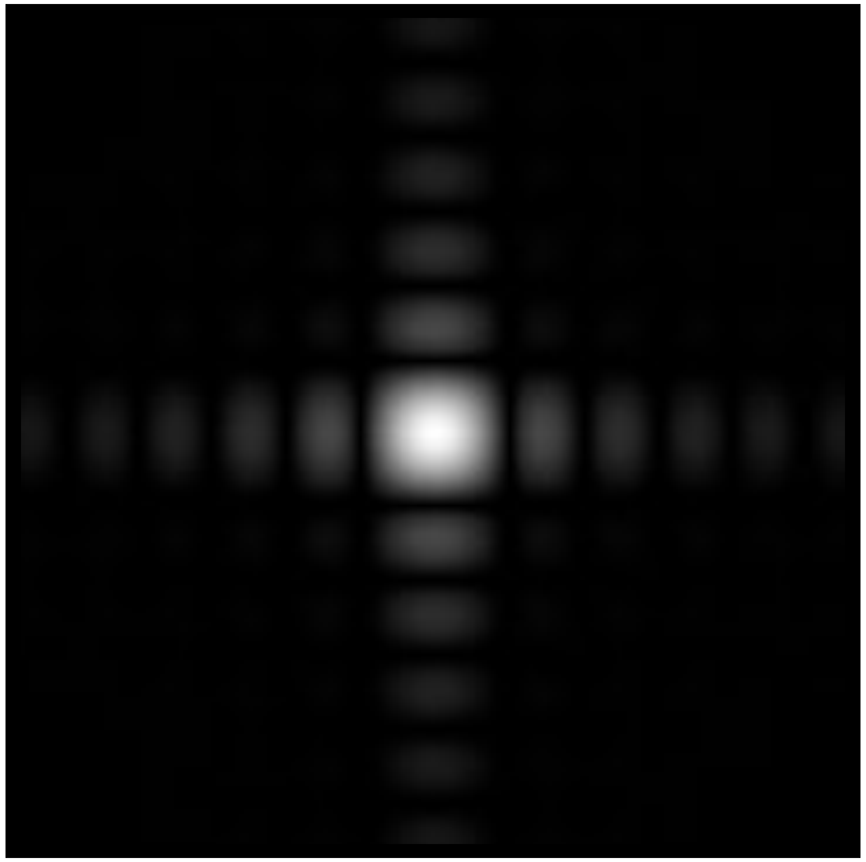
Note antialiased edges in rasterized triangle where pixel values take intermediate values

A Practical Pre-Filter

A 1 pixel-width box filter (low pass, blurring)



Spatial Domain



Frequency Domain

Antialiasing By Averaging Values in Pixel Area

Solution:

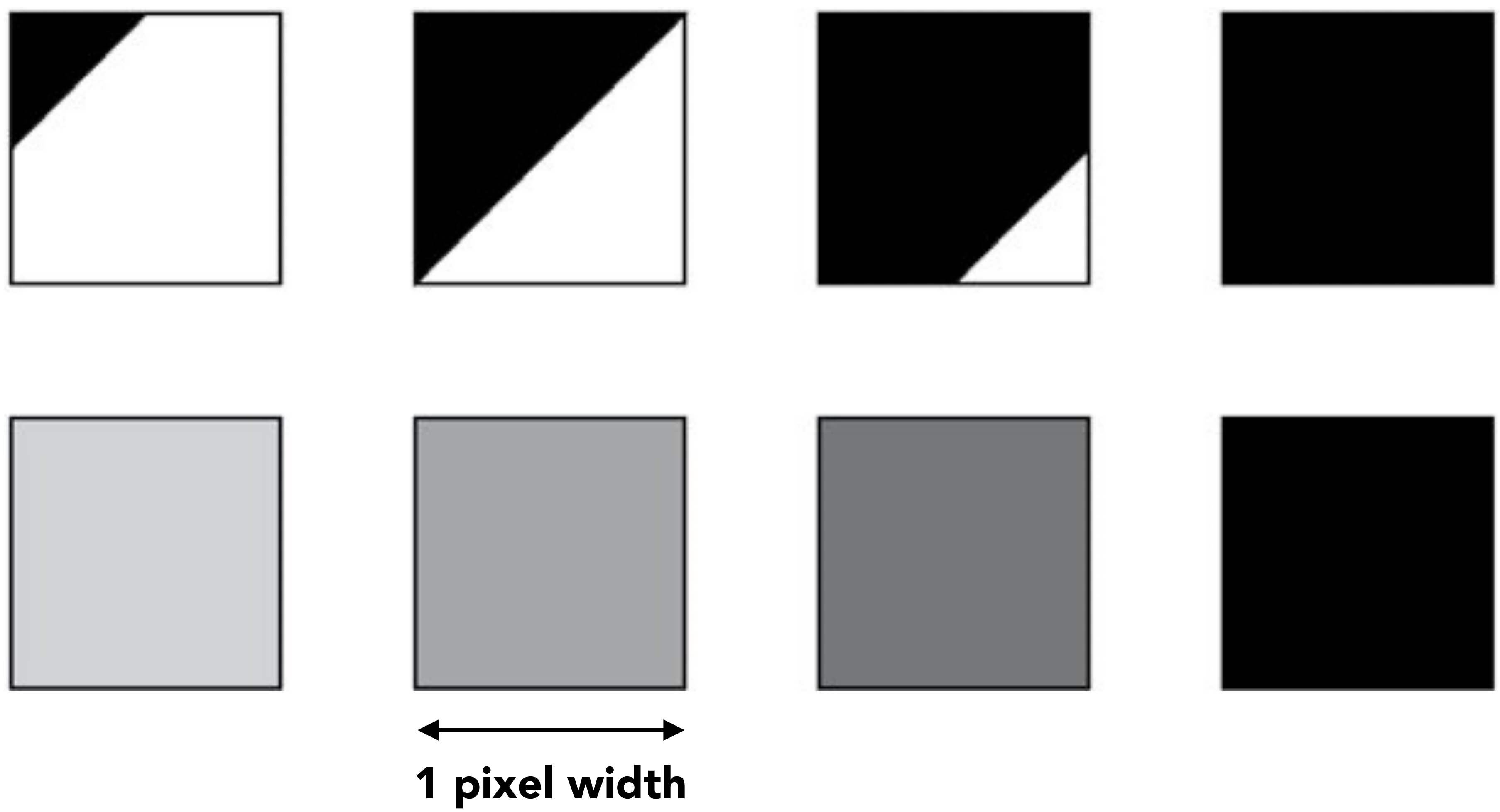
- **Convolve** f(x,y) by a 1-pixel box-blur
  - Recall: convolving = filtering = averaging
- **Then sample** at every pixel's center

Antialiasing by Computing Average Pixel Value

In rasterizing one triangle, the average value inside a pixel area of f(x,y) = inside(triangle,x,y) is equal to the area of the pixel covered by the triangle.

Original

Filtered

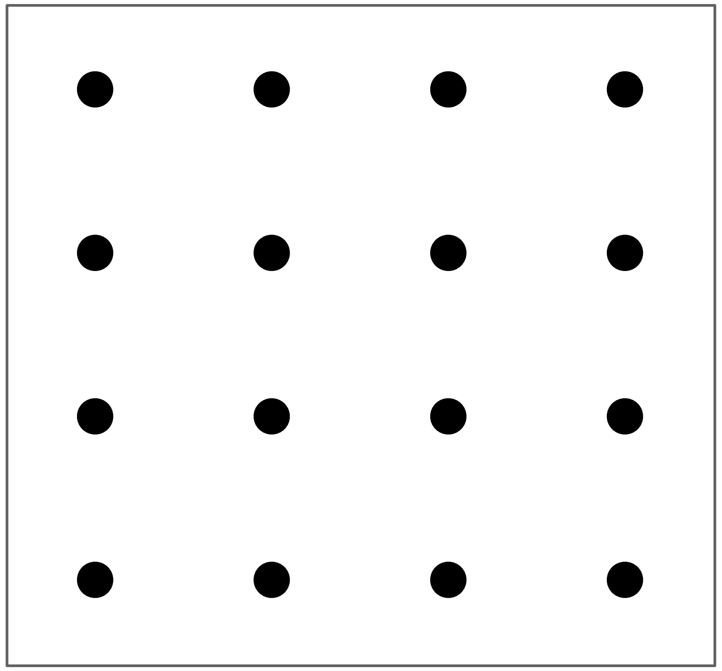


Antialiasing By Supersampling (MSAA)

Supersampling

Approximate the effect of the 1-pixel box filter by sampling multiple locations within a pixel and averaging their values:

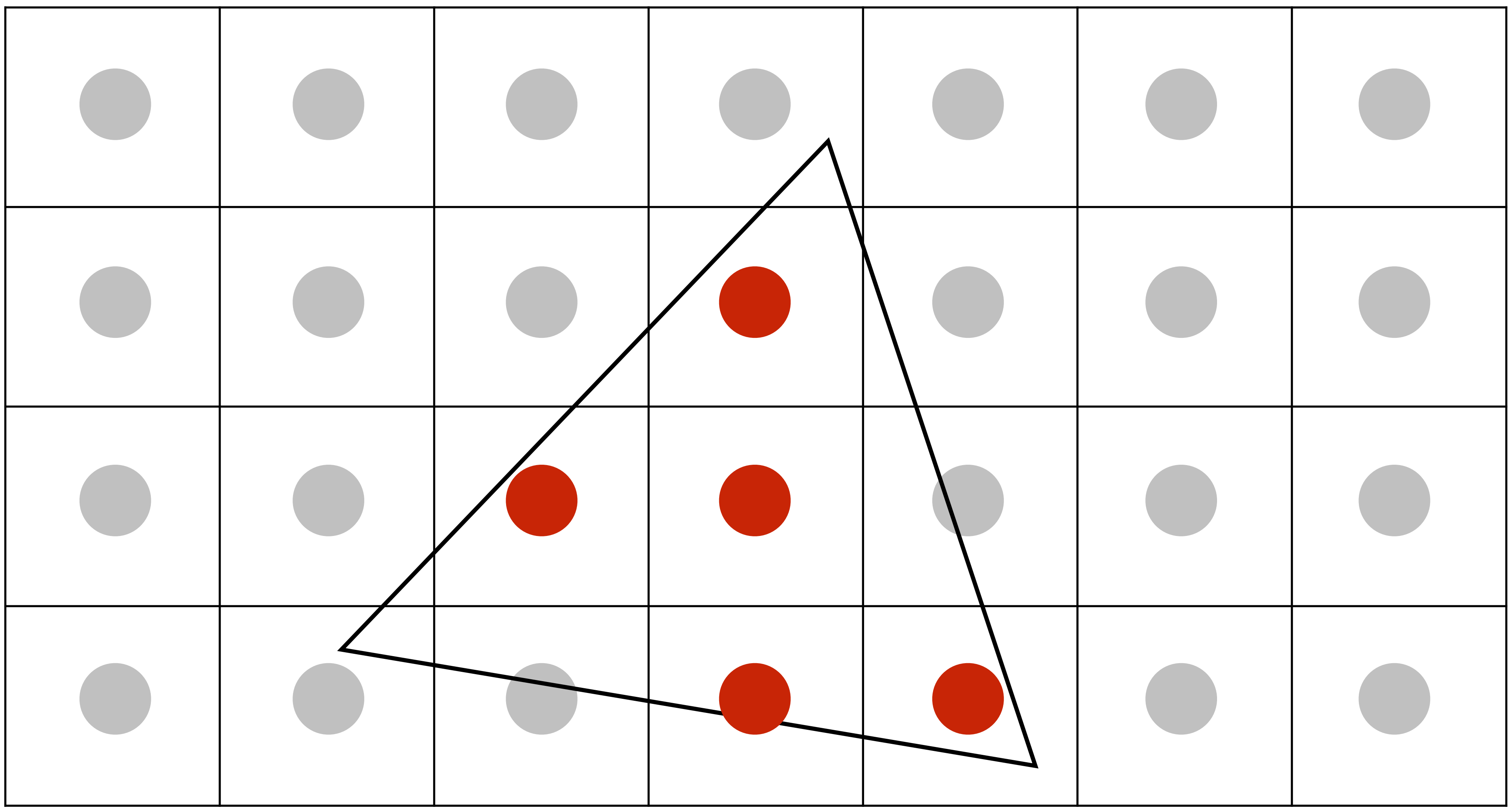
假设把一个像素分成4*4份，判断每一份有无在三角形内，把判断结果进行平均，判断结果相当于三角形对这个像素覆盖区域的一个近似



4x4 supersampling

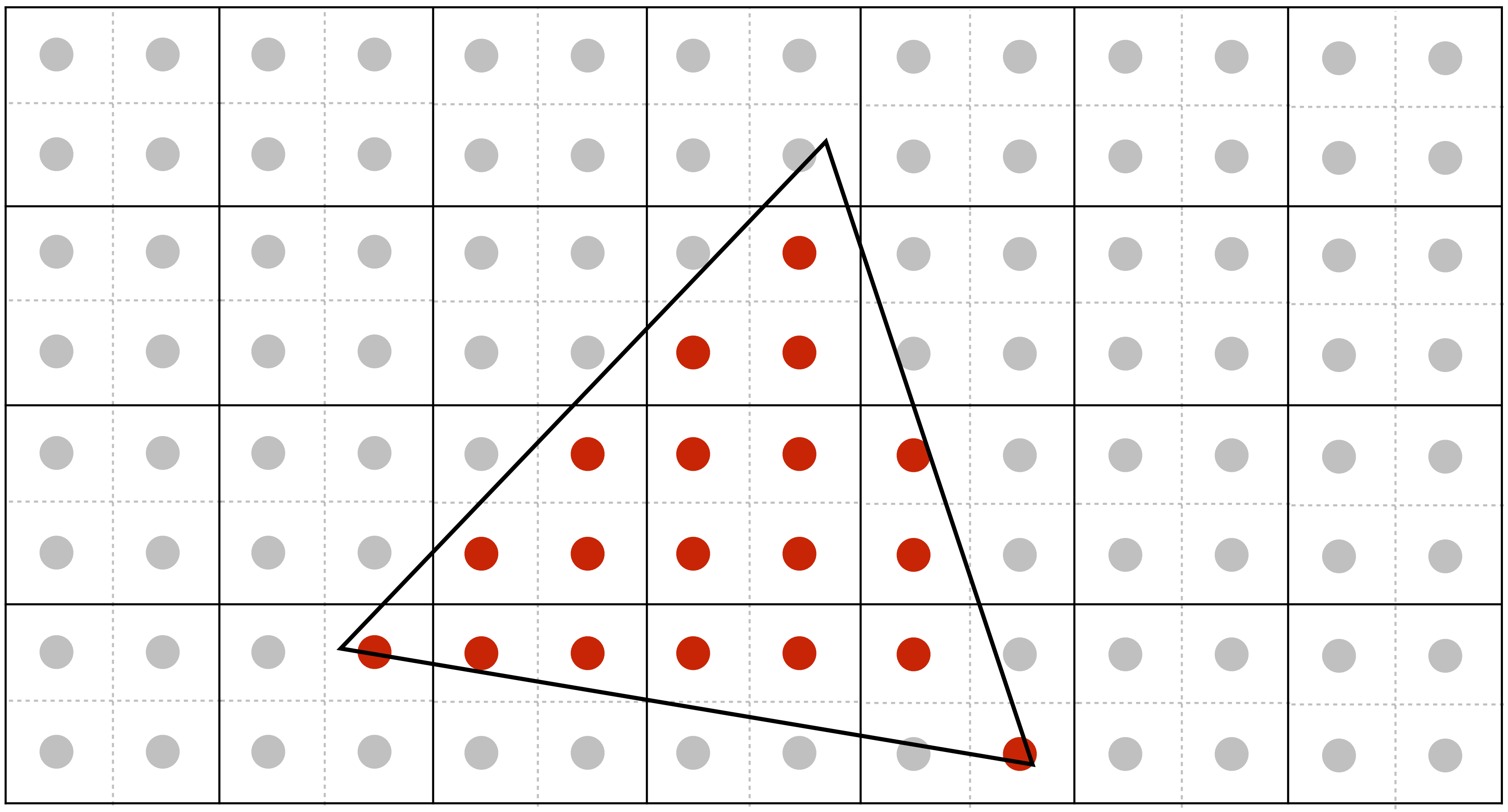
假设用更多的点，用100*100的点，会得到更准确的近似

Point Sampling: One Sample Per Pixel



Supersampling: Step 1

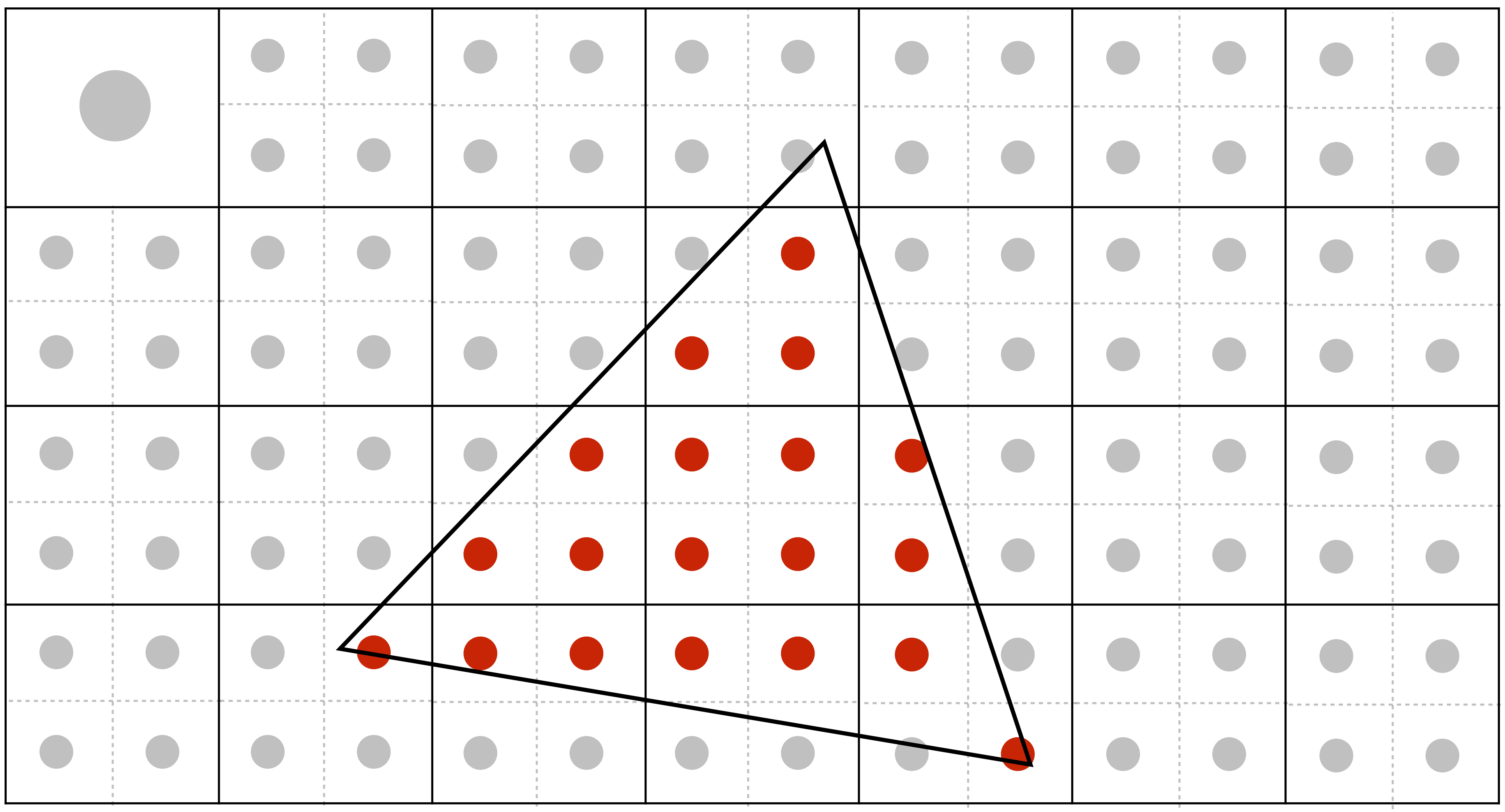
Take NxN samples in each pixel.



2x2 supersampling

Supersampling: Step 2

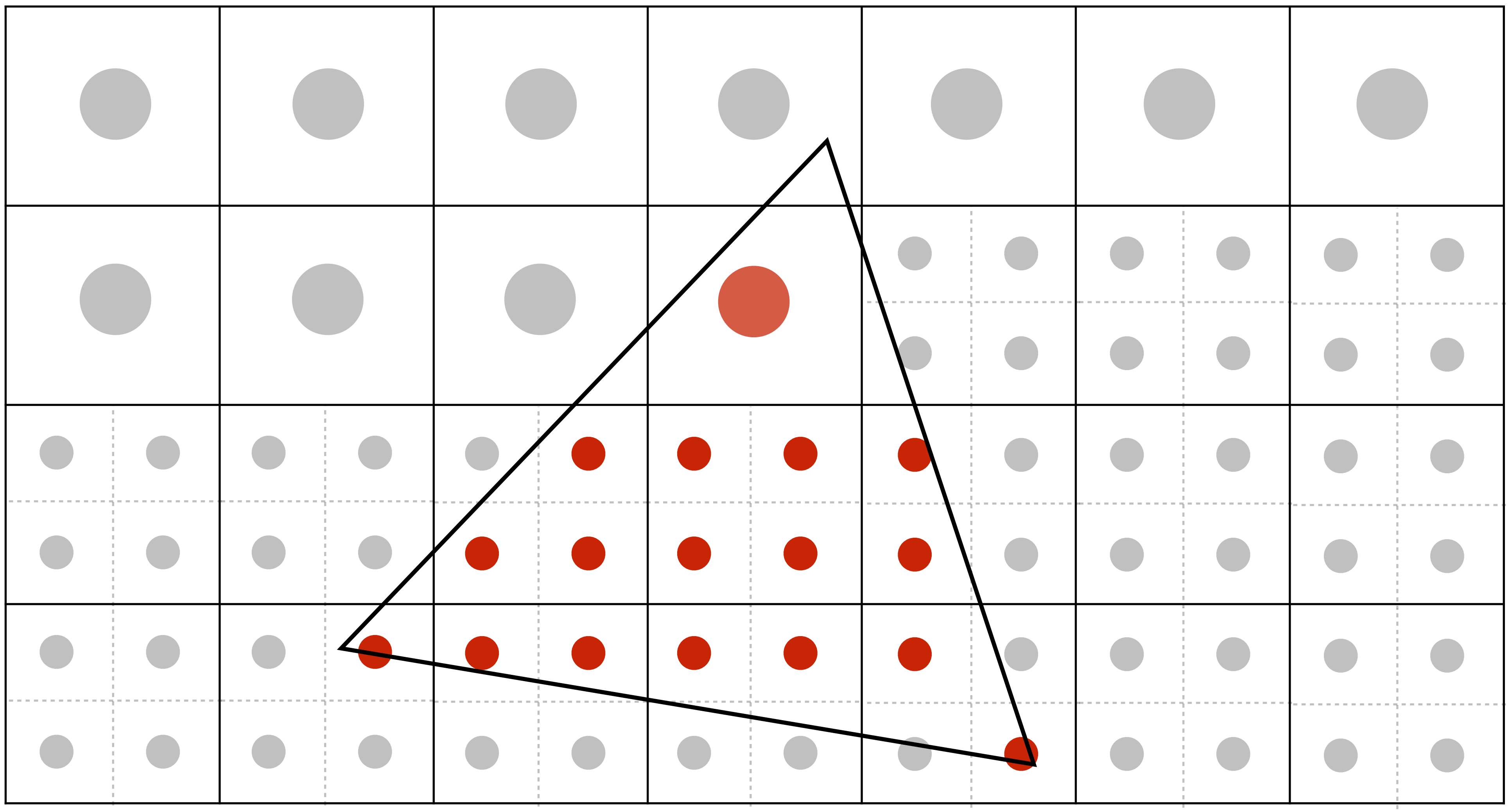
Average the NxN samples “inside” each pixel.



Averaging down

Supersampling: Step 2

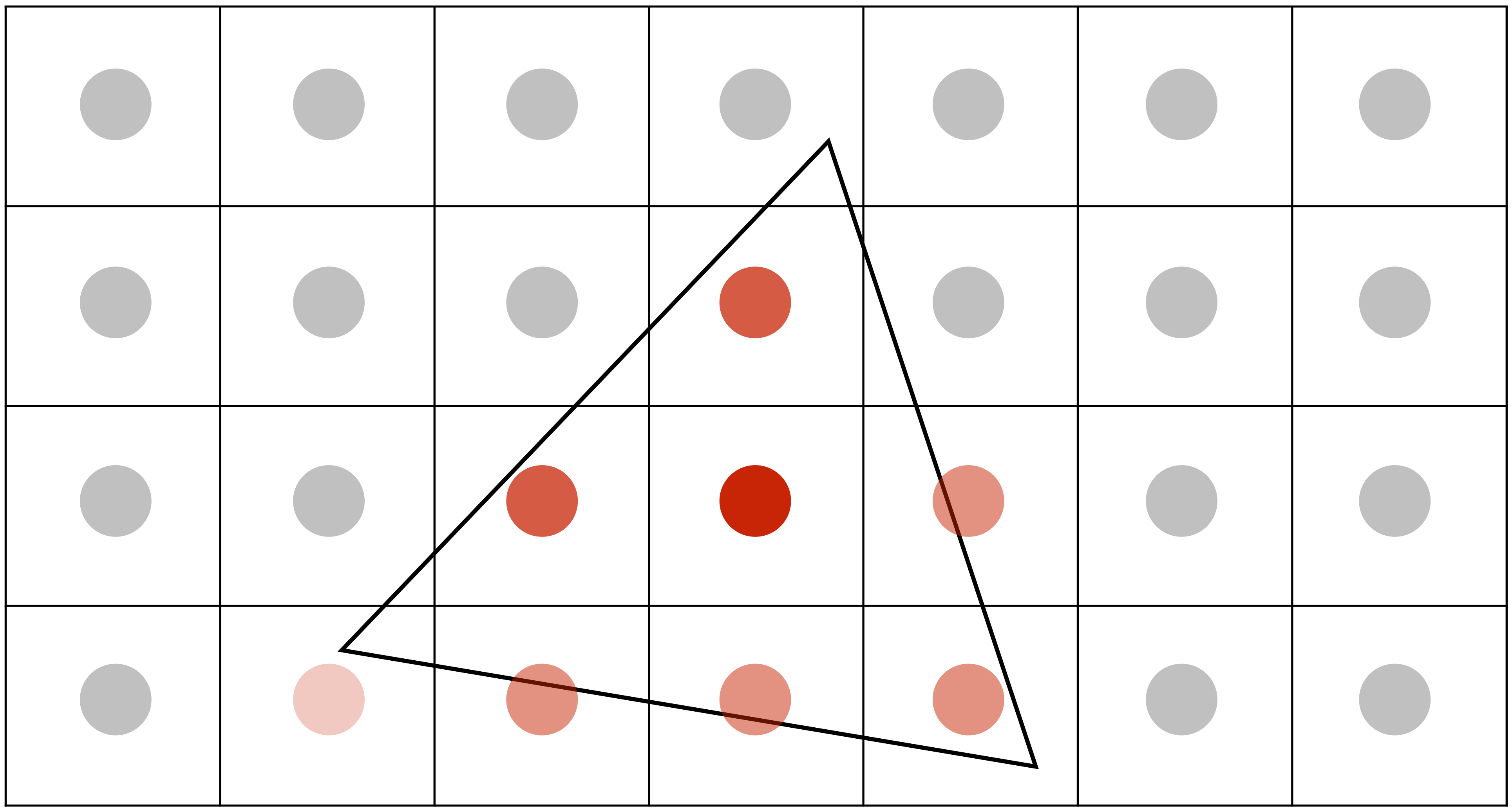
Average the NxN samples “inside” each pixel.



Averaging down

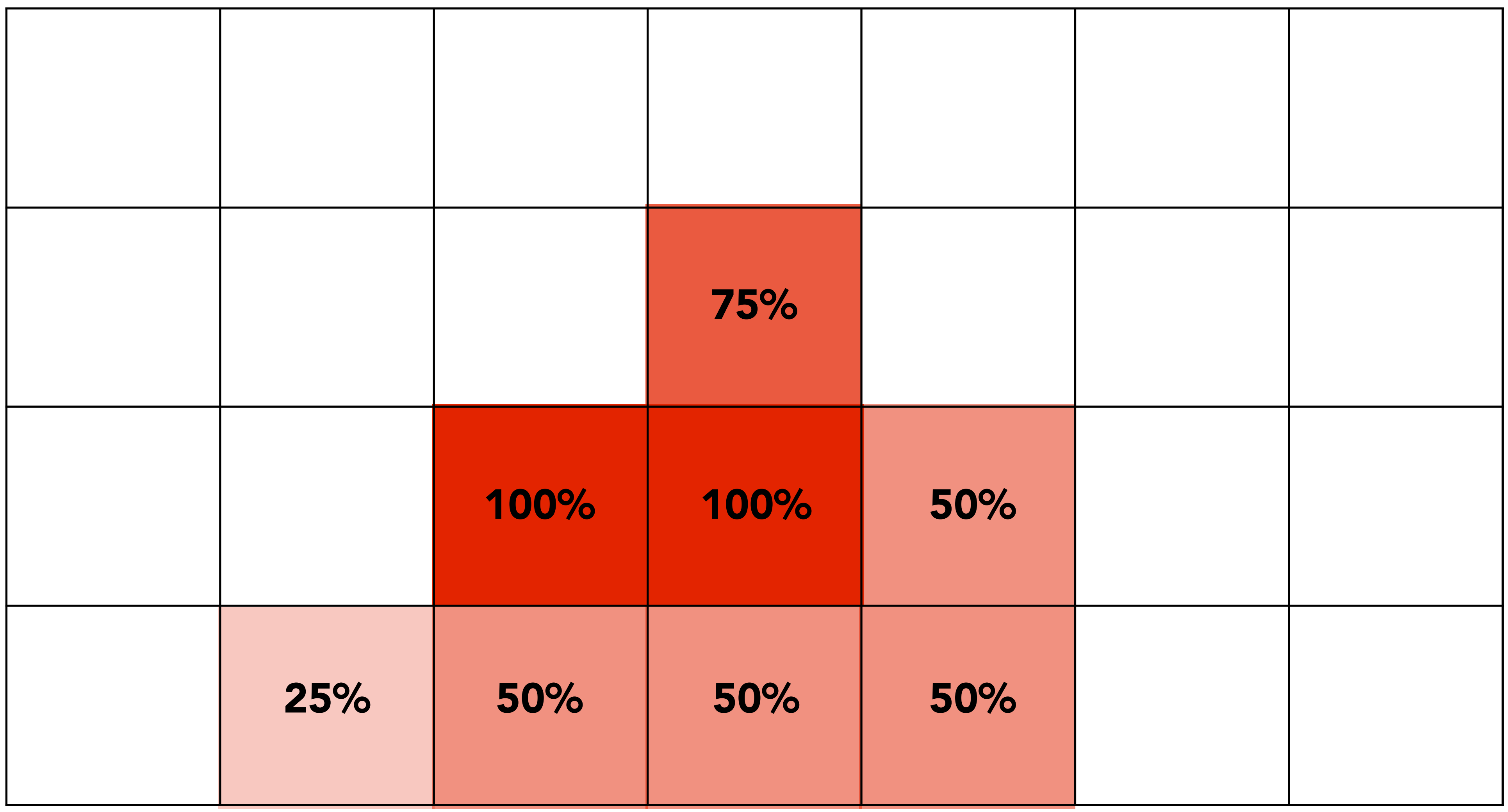
Supersampling: Step 2

Average the NxN samples "inside" each pixel.



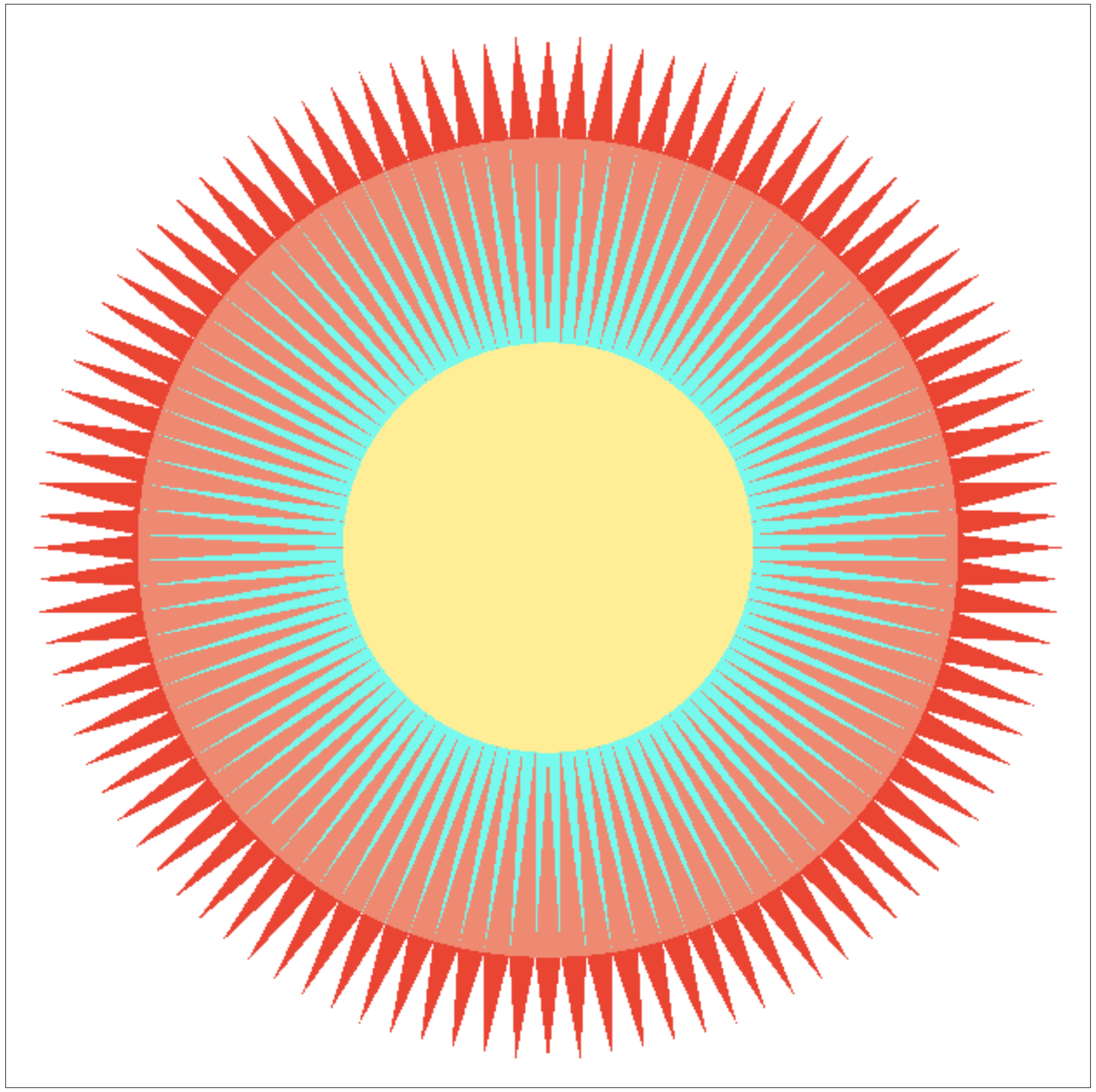
Supersampling: Result

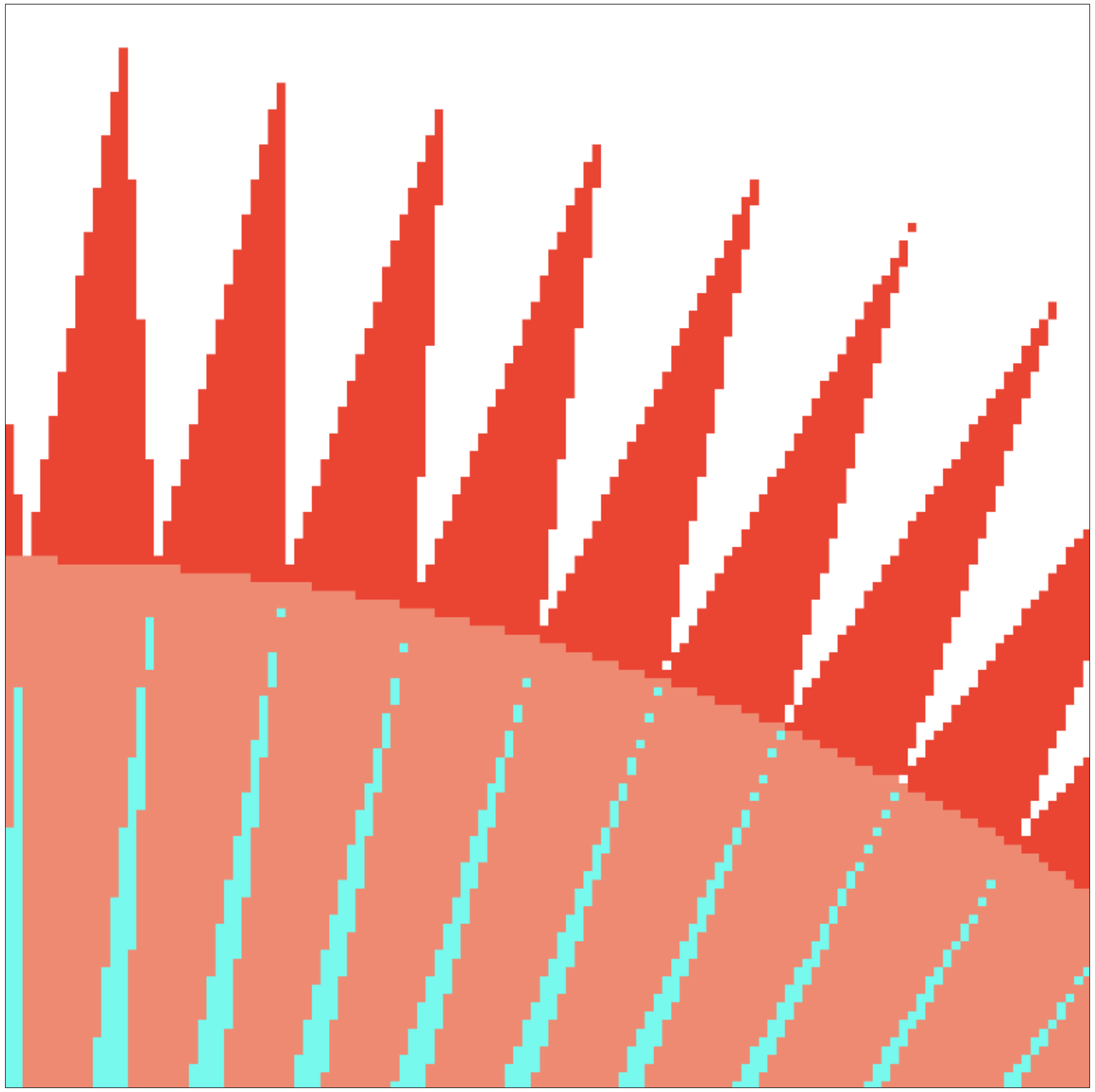
This is the corresponding signal emitted by the display



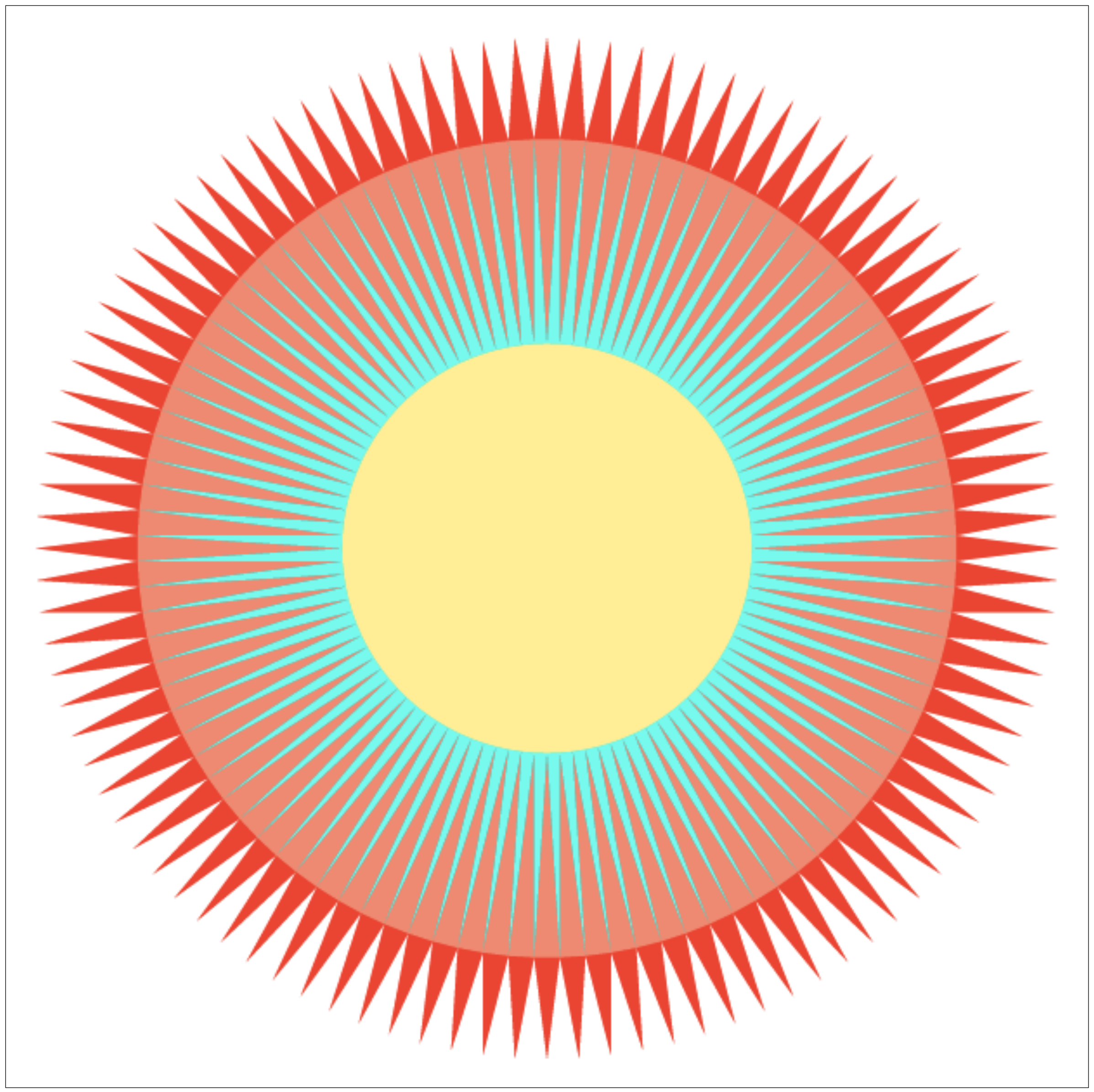
对一个像素增加采样次数，对像素内的每个采样点是否被覆盖相加平均，像素被三角形覆盖的比例

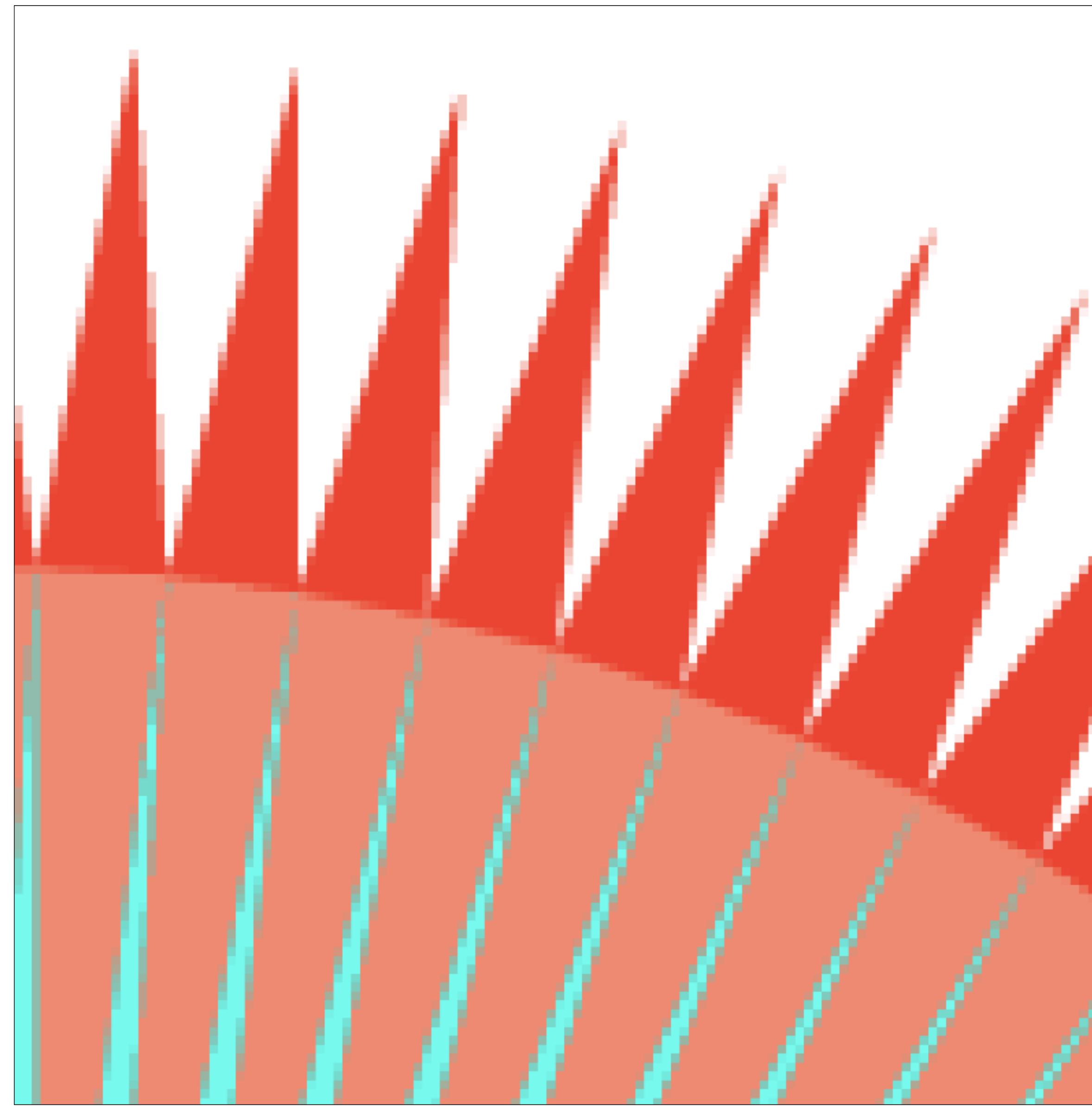
Point Sampling





4x4 Supersampling





Antialiasing Today

工业中利用不规则的图案，多个像素会复用

No free lunch!

- What's the cost of MSAA? 增大了计算量，使用2*2增加4被计算量，4*4增加16倍计算量

Milestones (personal idea)

- FXAA (Fast Approximate AA) 相当于后期的图像处理 得到一幅图找到边界，去除走样的点
- TAA (Temporal AA) 时间的 找上一帧的画面

Super resolution / super sampling

- From low resolution to high resolution
- Essentially still "not enough samples" problem
- DLSS (Deep Learning Super Sampling)

Thank you!