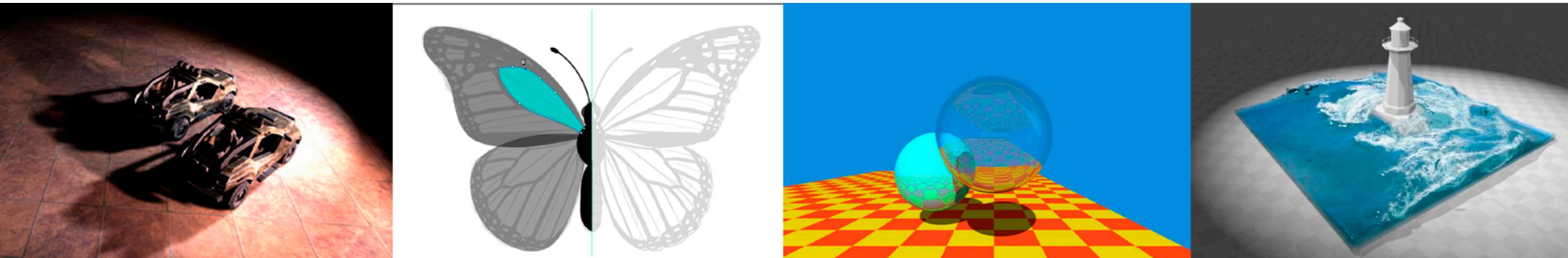


# Introduction to Computer Graphics

GAMES101, Lingqi Yan, UC Santa Barbara

反走样与深度缓冲

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



# 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

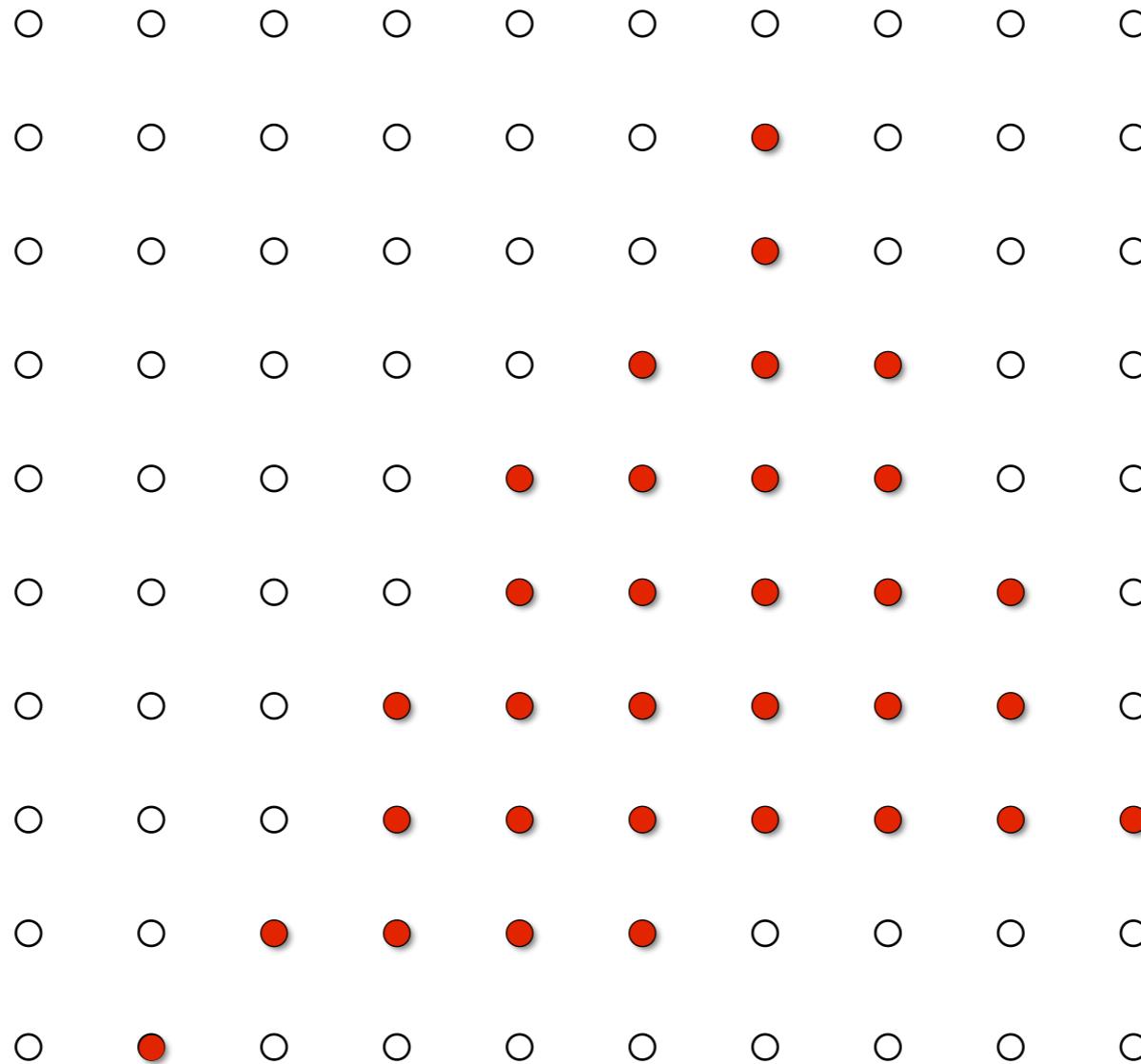
# Last Lectures

- Viewing  
- View + Projection + Viewport  
    视口变换
- Rasterizing triangles  
    三角光栅化  
- Point-in-triangle test  
- Aliasing

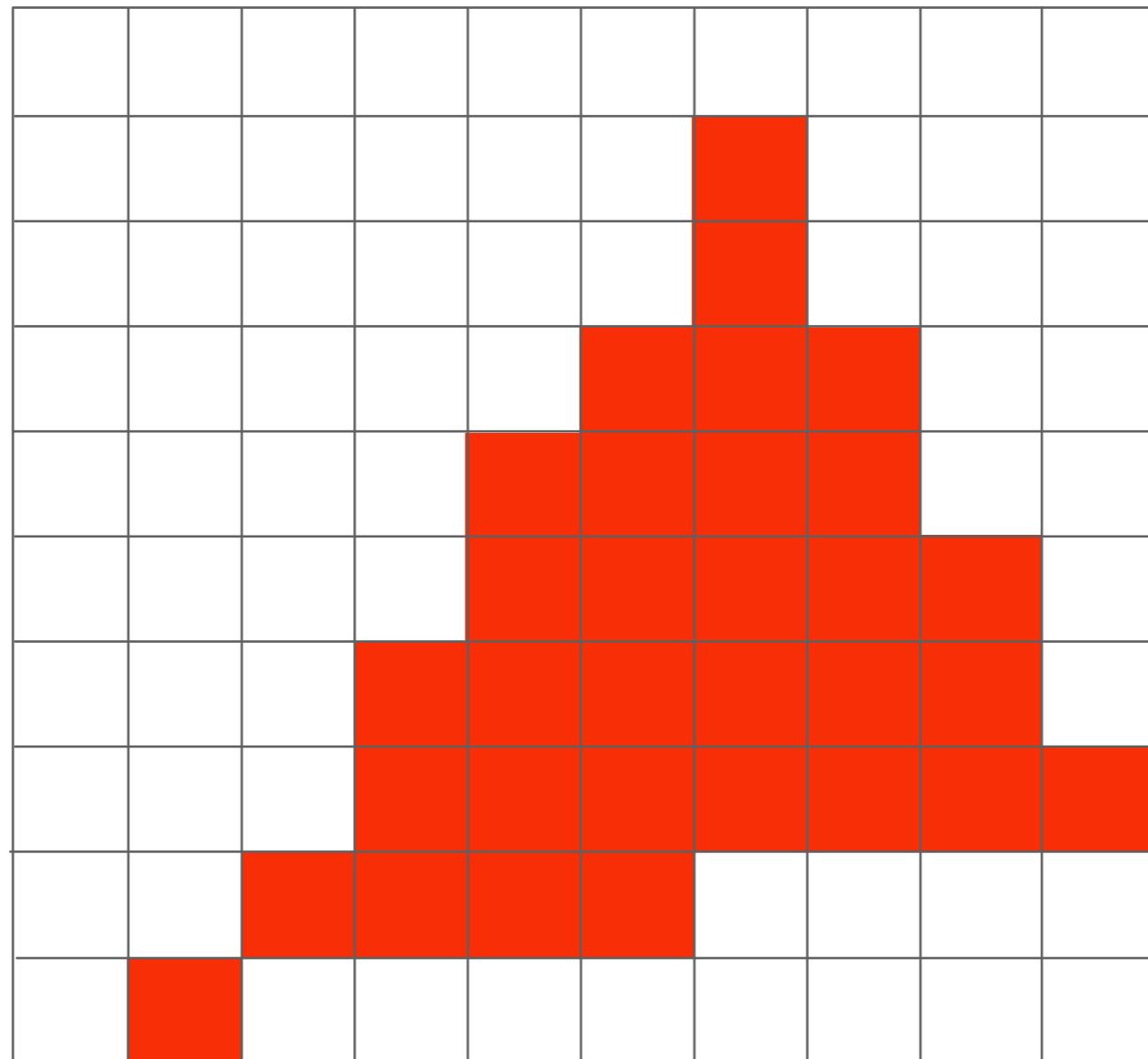
# Today

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

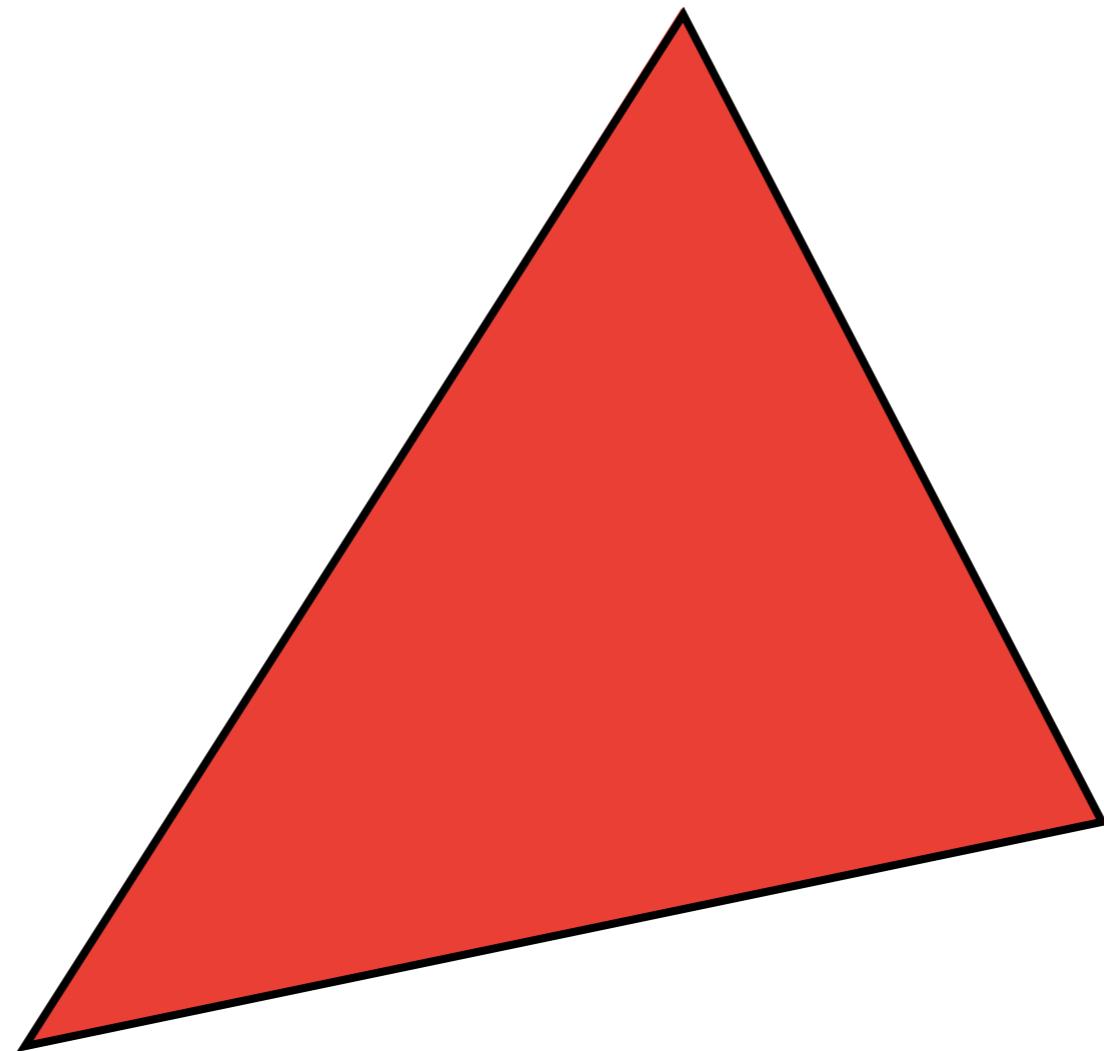
# Recap: Testing in/out $\Delta$ at pixels' centers



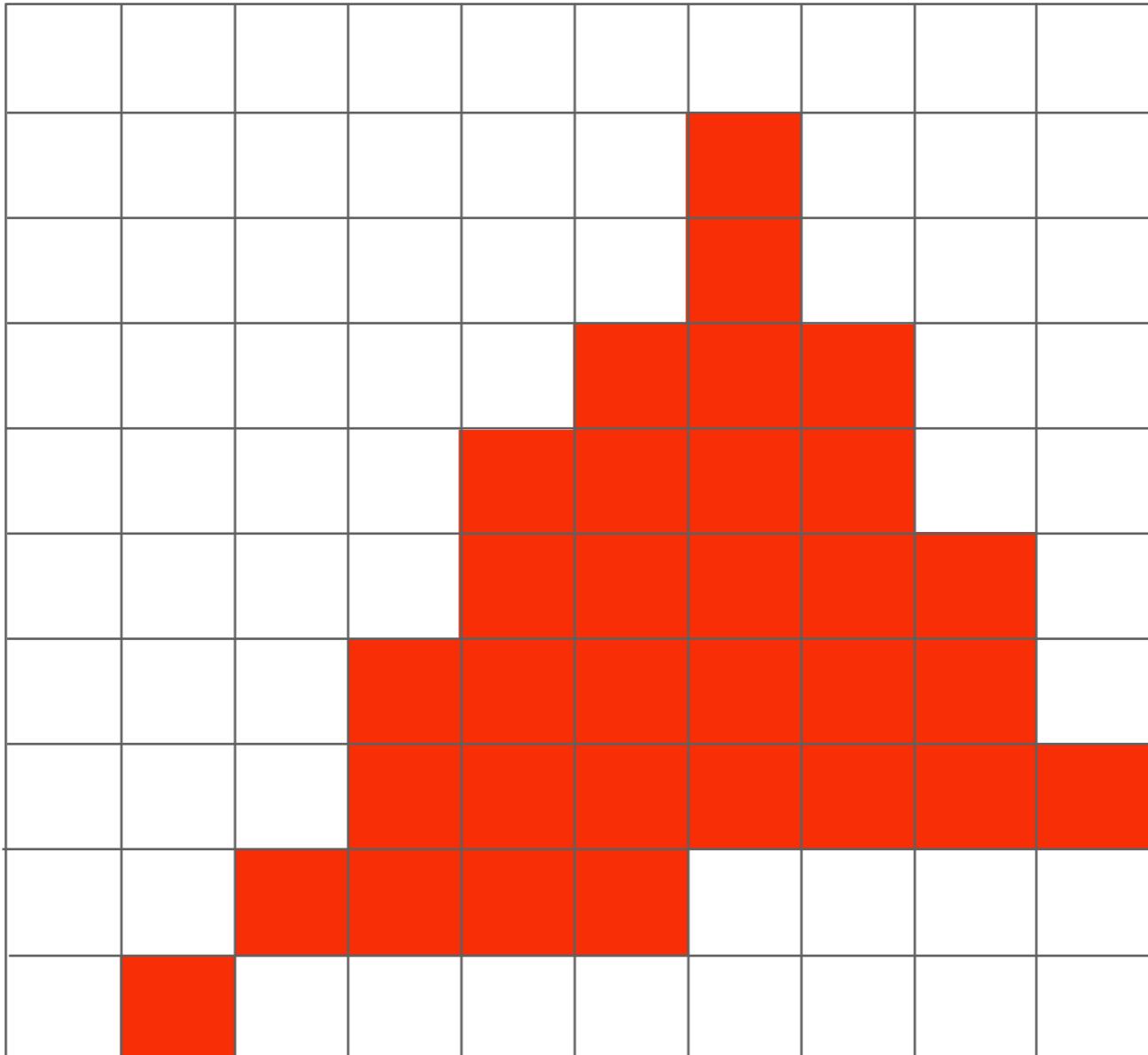
# Pixels are uniformly-colored squares



# Compare: The Continuous Triangle Function

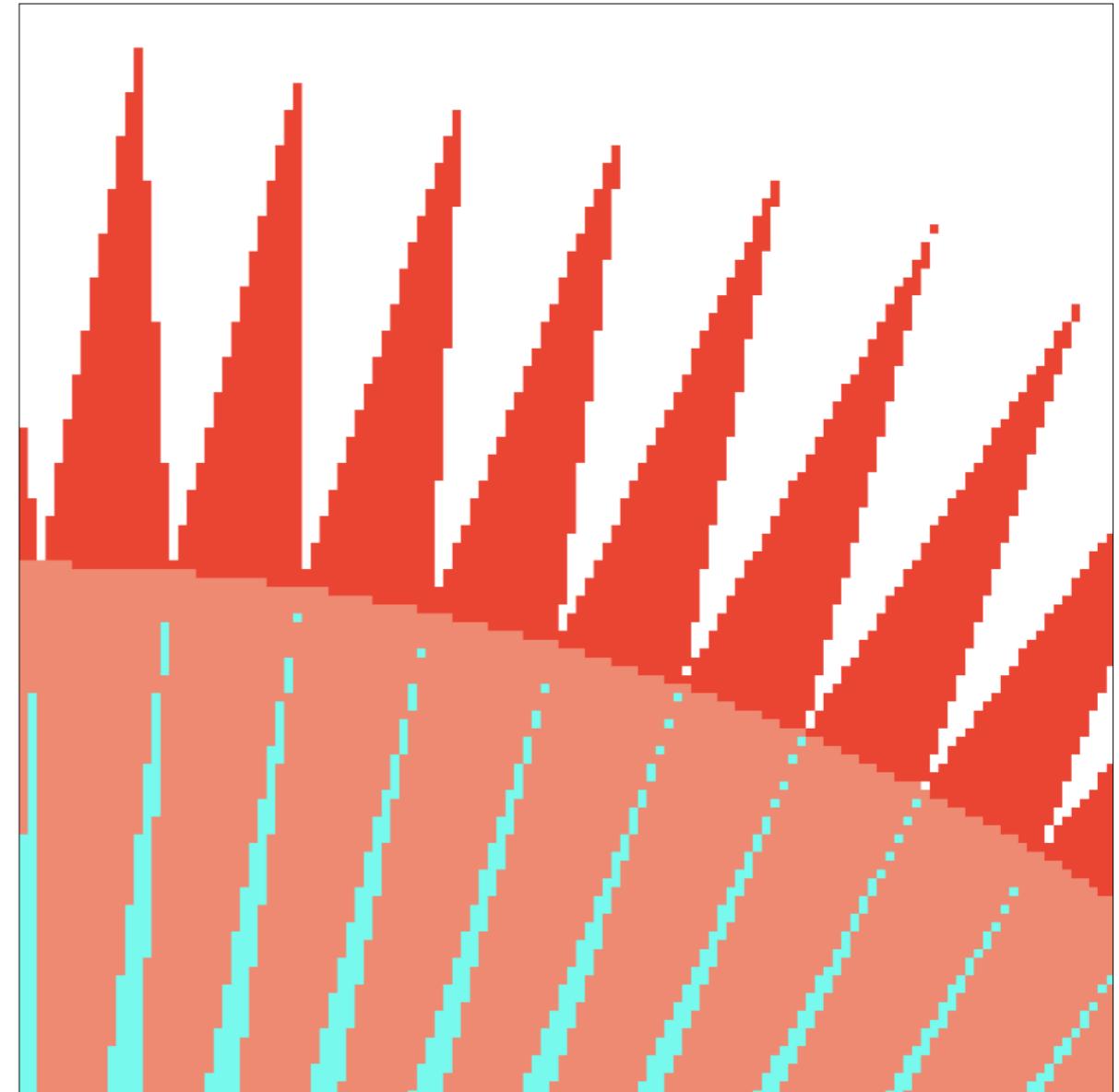
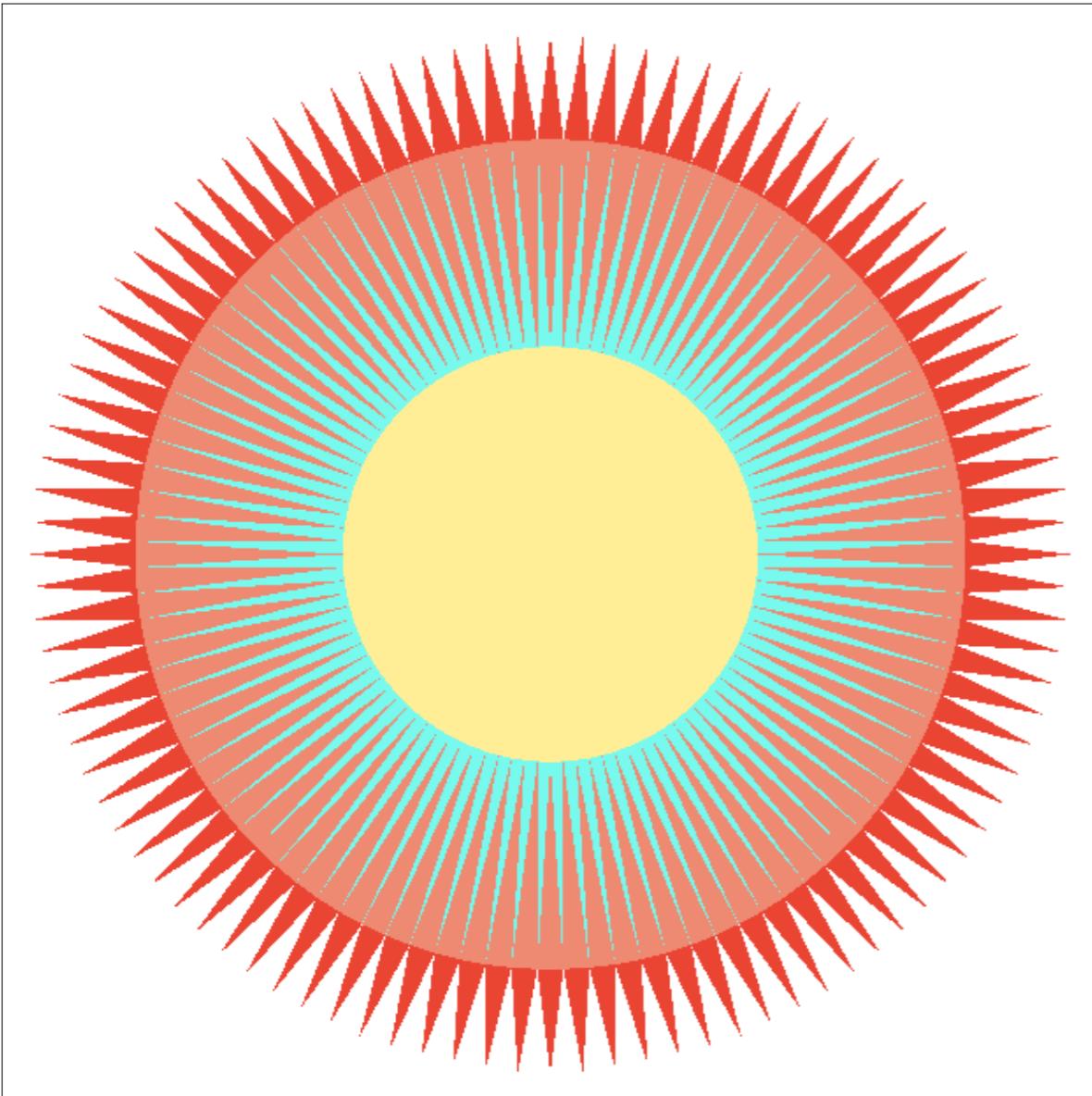


# What's Wrong With This Picture?



Jaggies!

# Aliasing



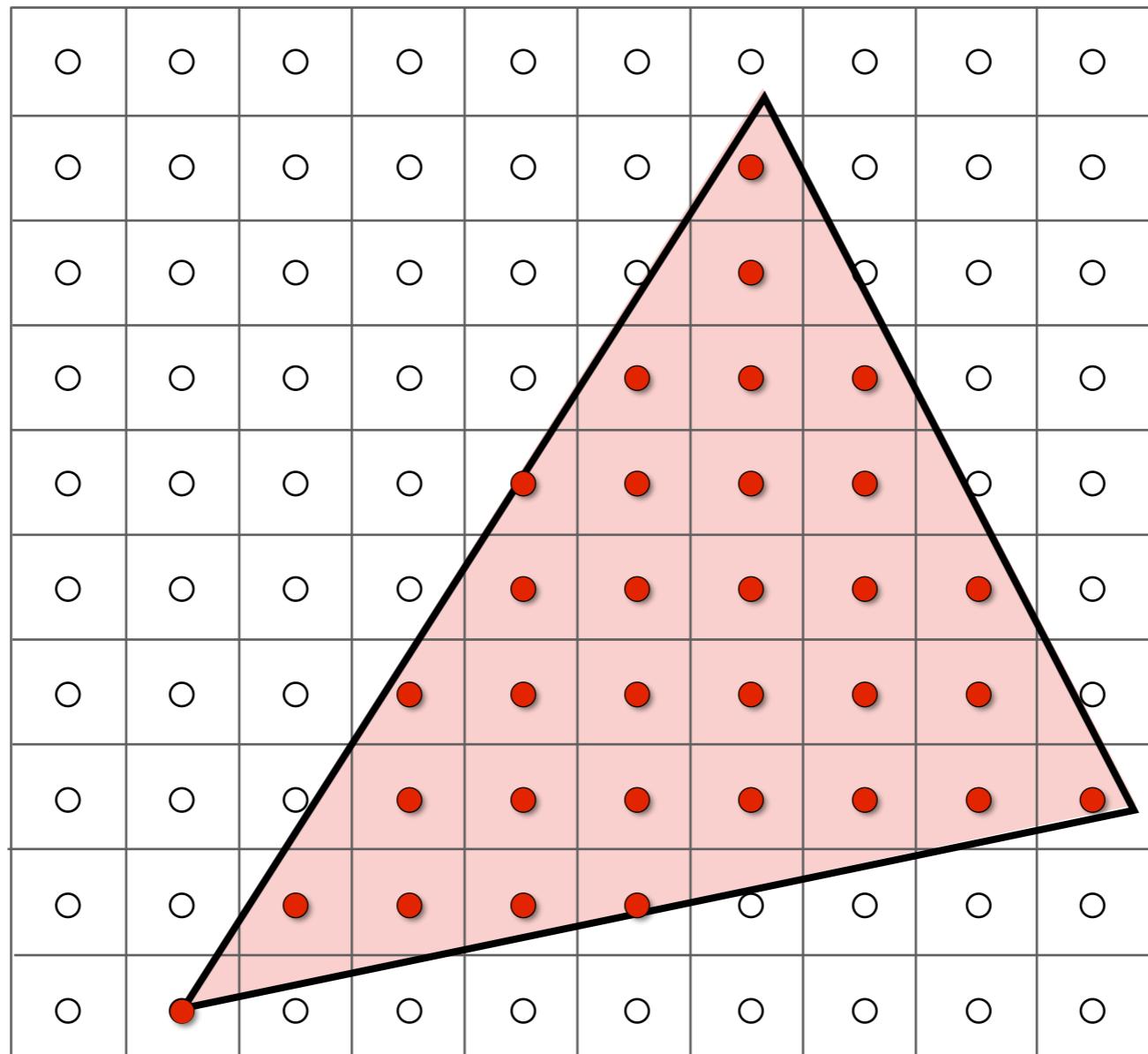
Is this the best we can do?

Slide courtesy of Prof. Ren Ng, UC Berkeley

Sampling is Ubiquitous in  
Computer Graphics

普遍存在

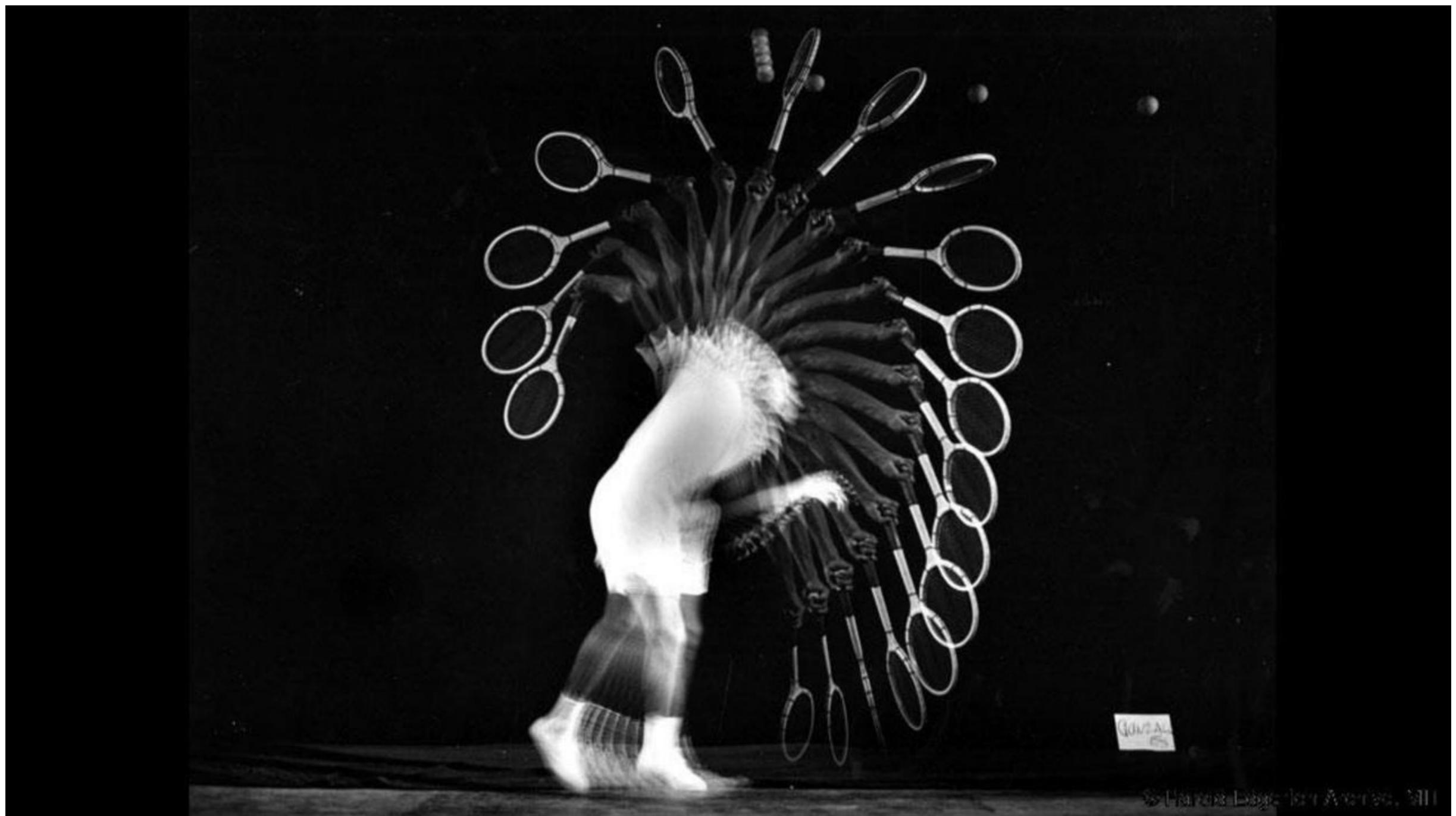
# Rasterization = Sample 2D Positions



# Photograph = Sample Image Sensor Plane



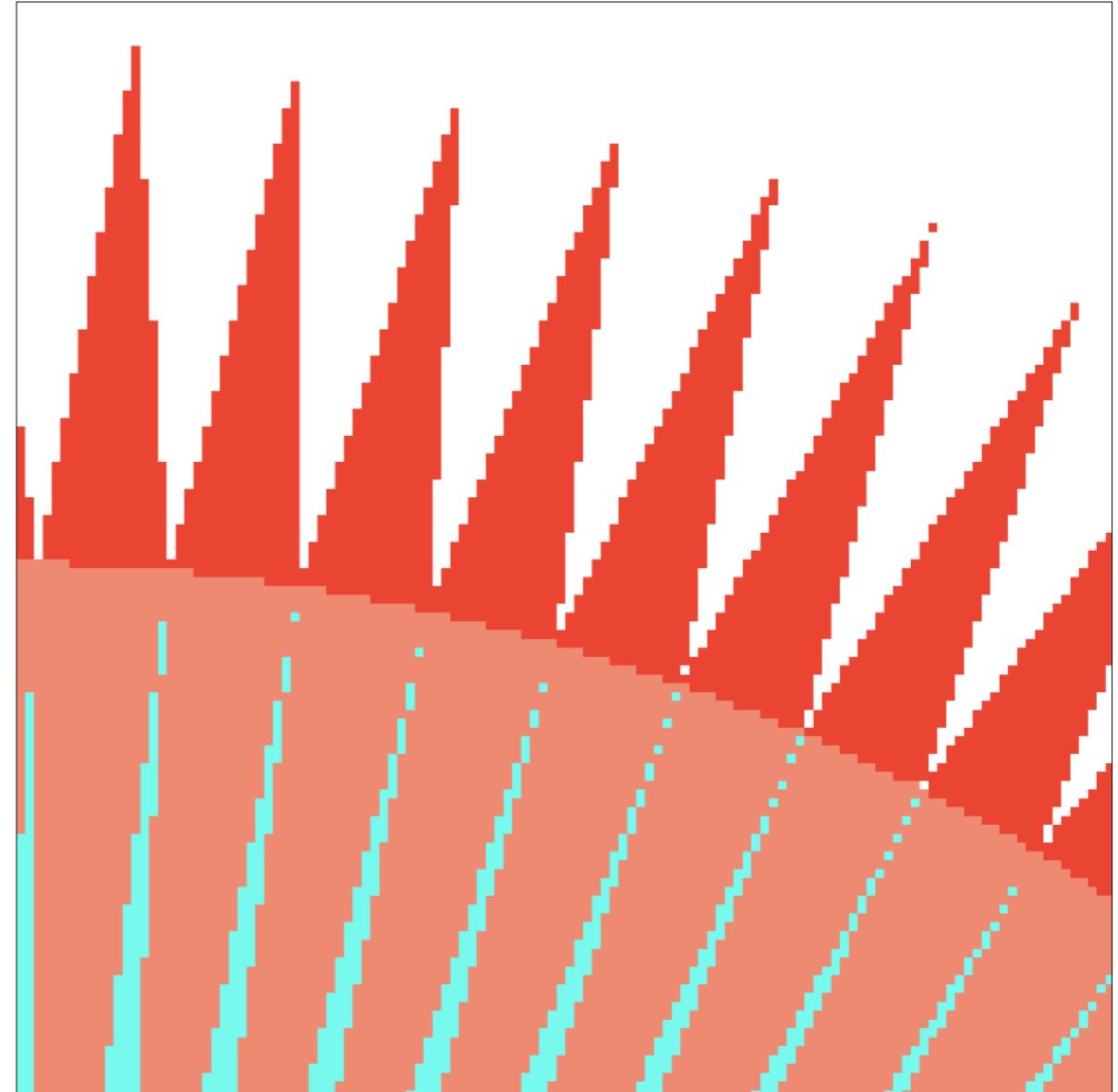
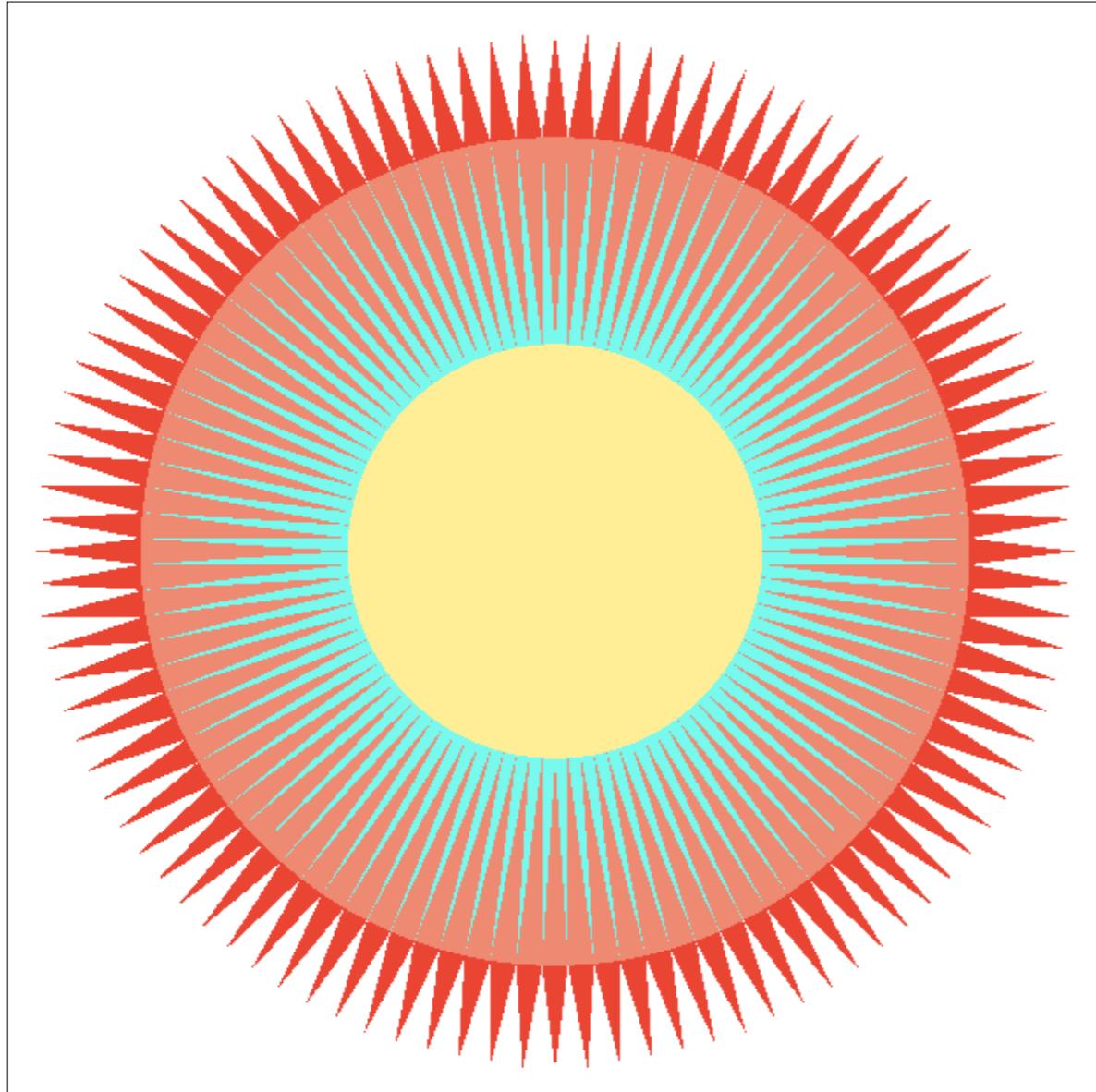
# Video = Sample Time



**Harold Edgerton Archive, MIT**

# 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

[mwa:]

摩尔纹

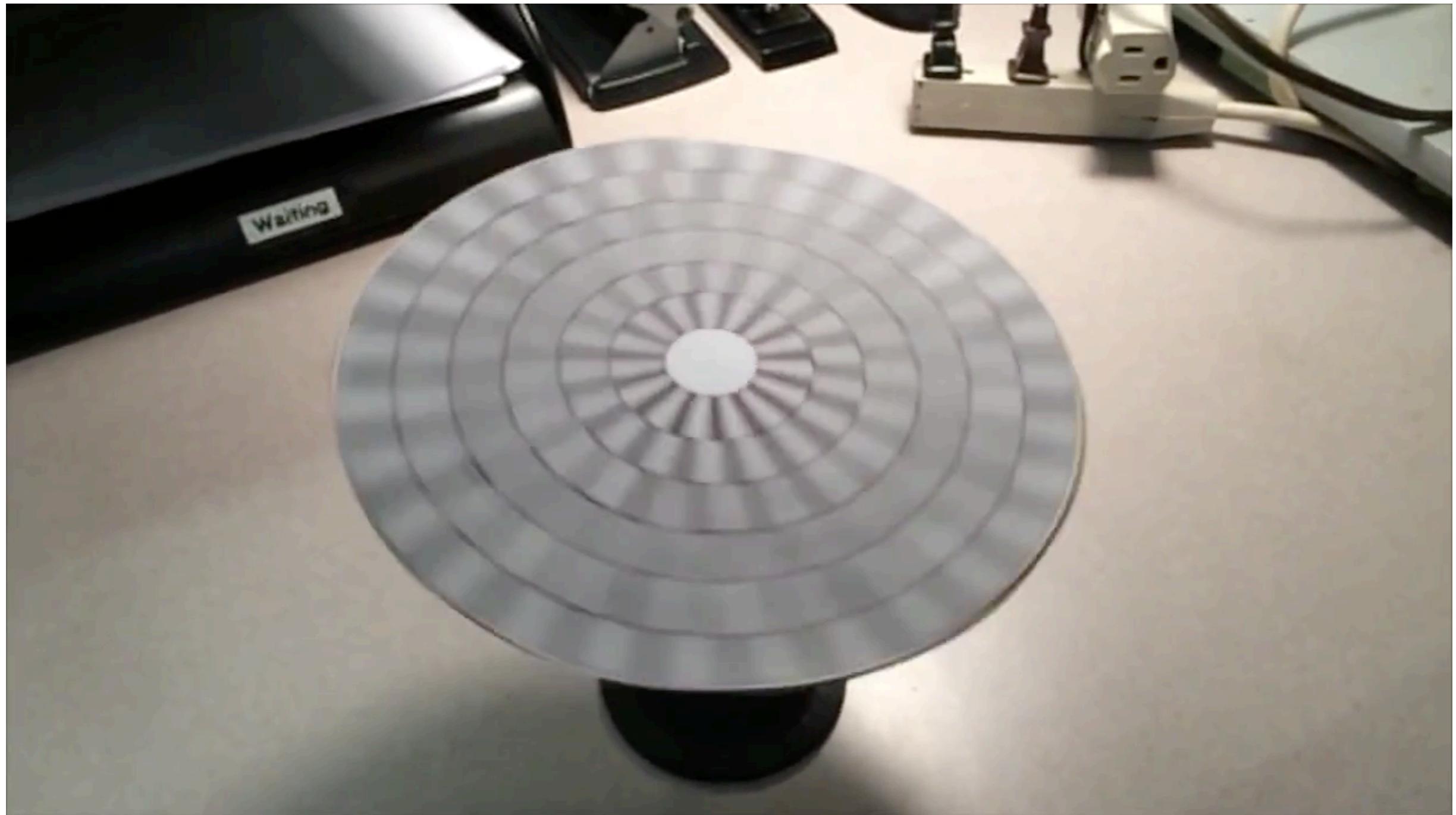


lystit.com

Skip odd rows and columns  
跳过奇数行与奇数列

人眼时间采样跟不上运动速度

# 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] ...

] 空间

时间

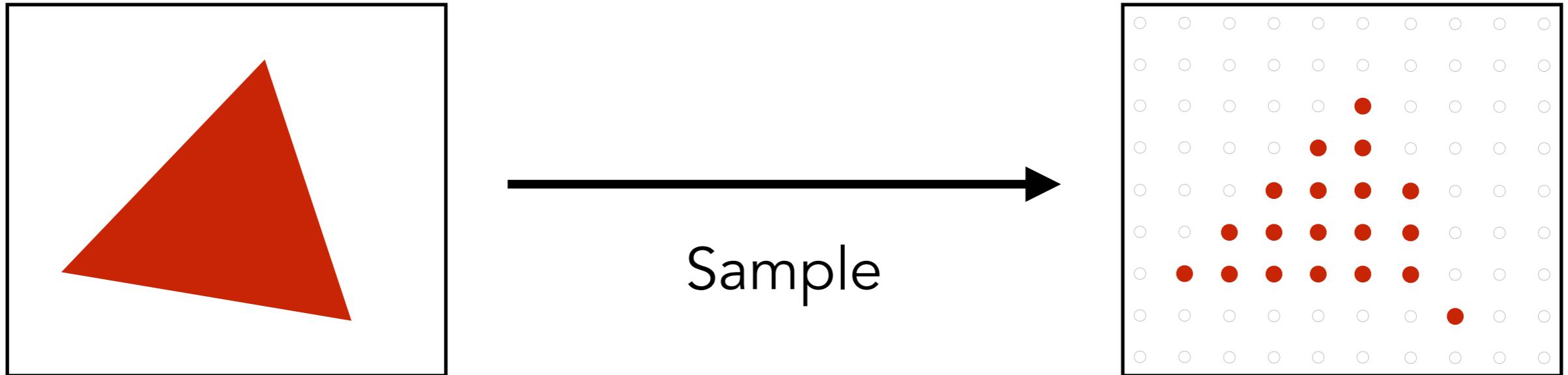
Behind the Aliasing Artifacts

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

Antialiasing Idea:  
Blurring (Pre-Filtering) Before  
Sampling

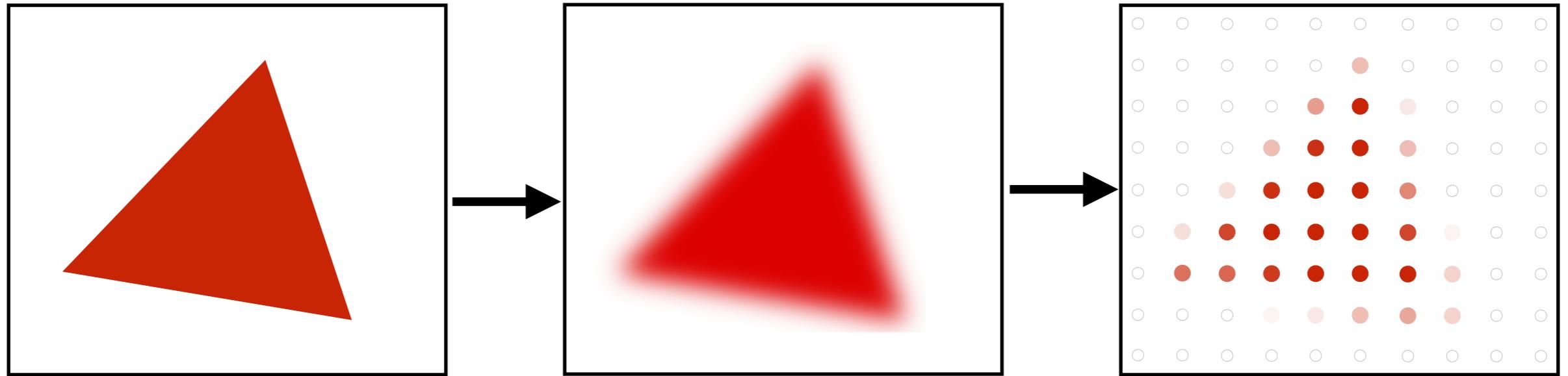
先模糊再采样

# Rasterization: Point Sampling in Space



Note jaggies in rasterized triangle  
where pixel values are **pure red or white**  
值变化太快

# Rasterization: Antialiased Sampling

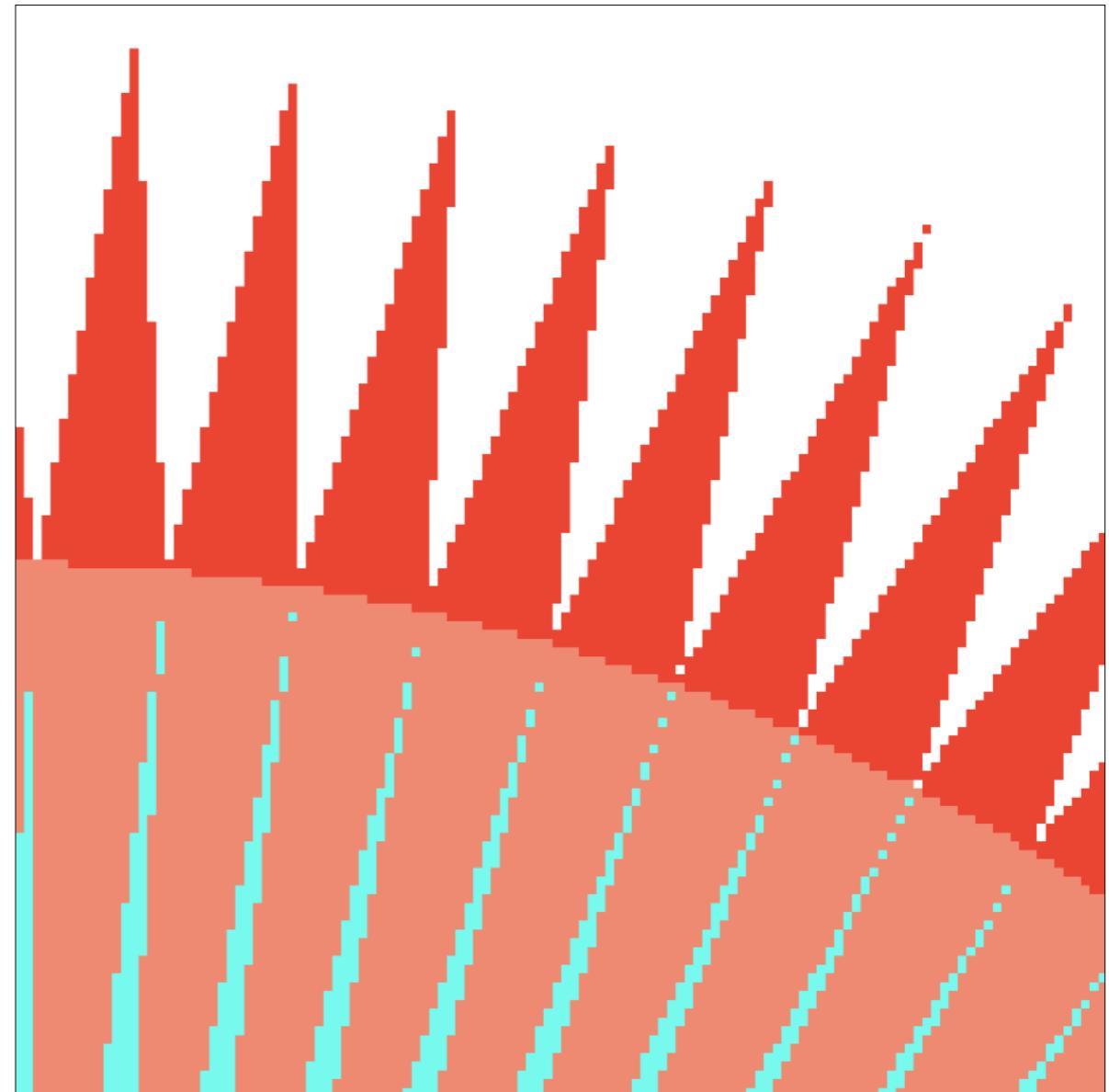
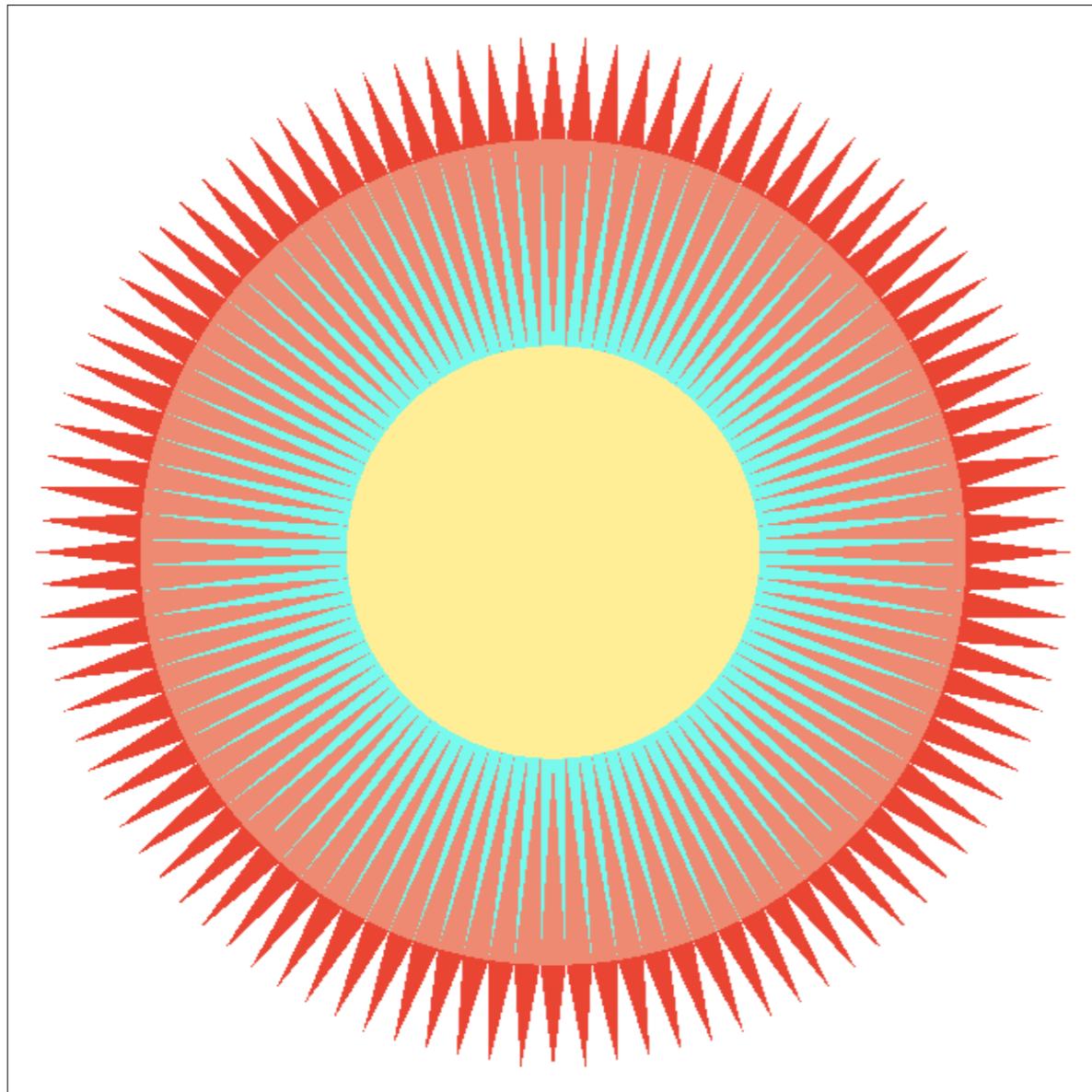


Pre-Filter  
(remove frequencies above Nyquist) (?)

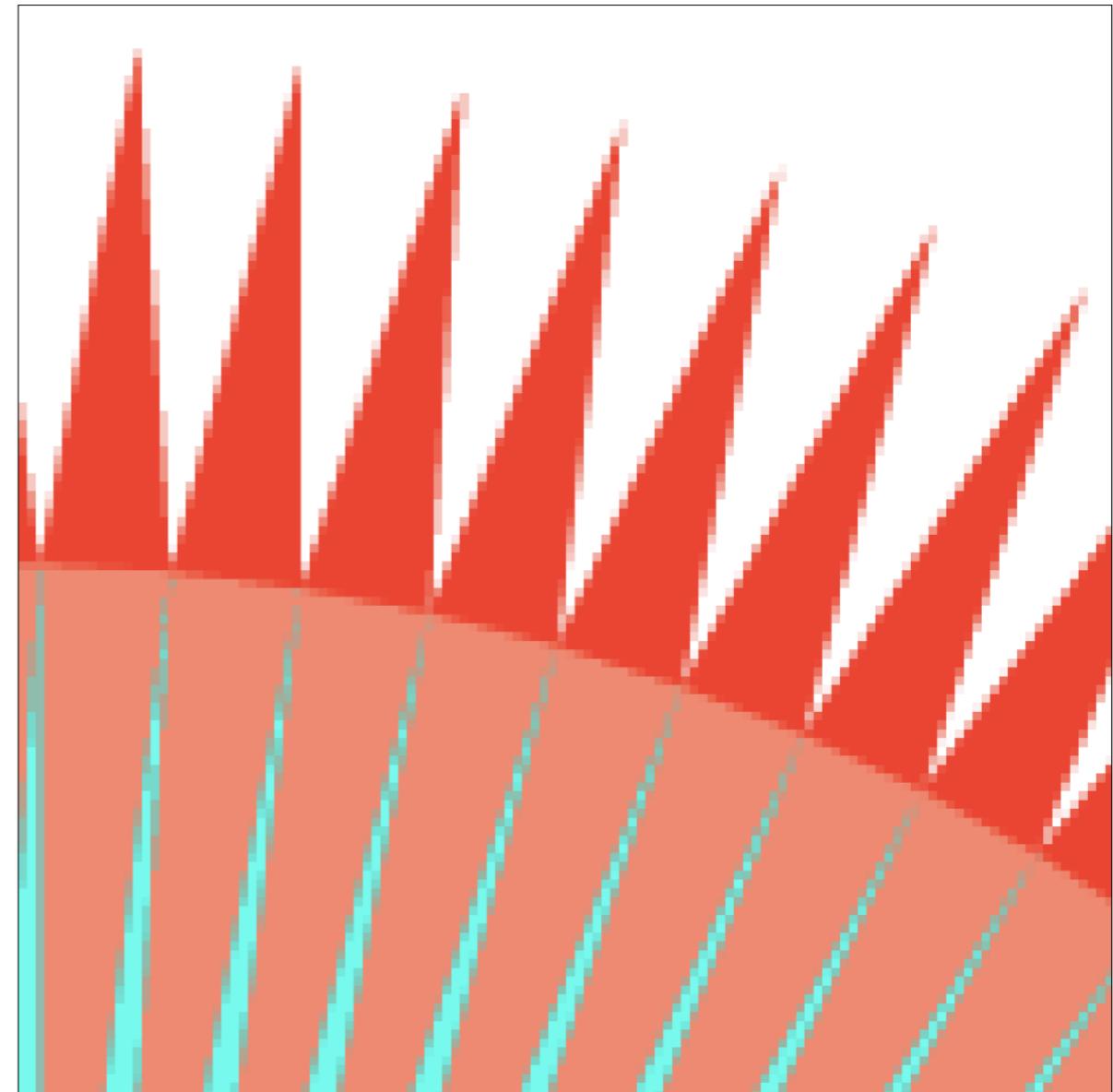
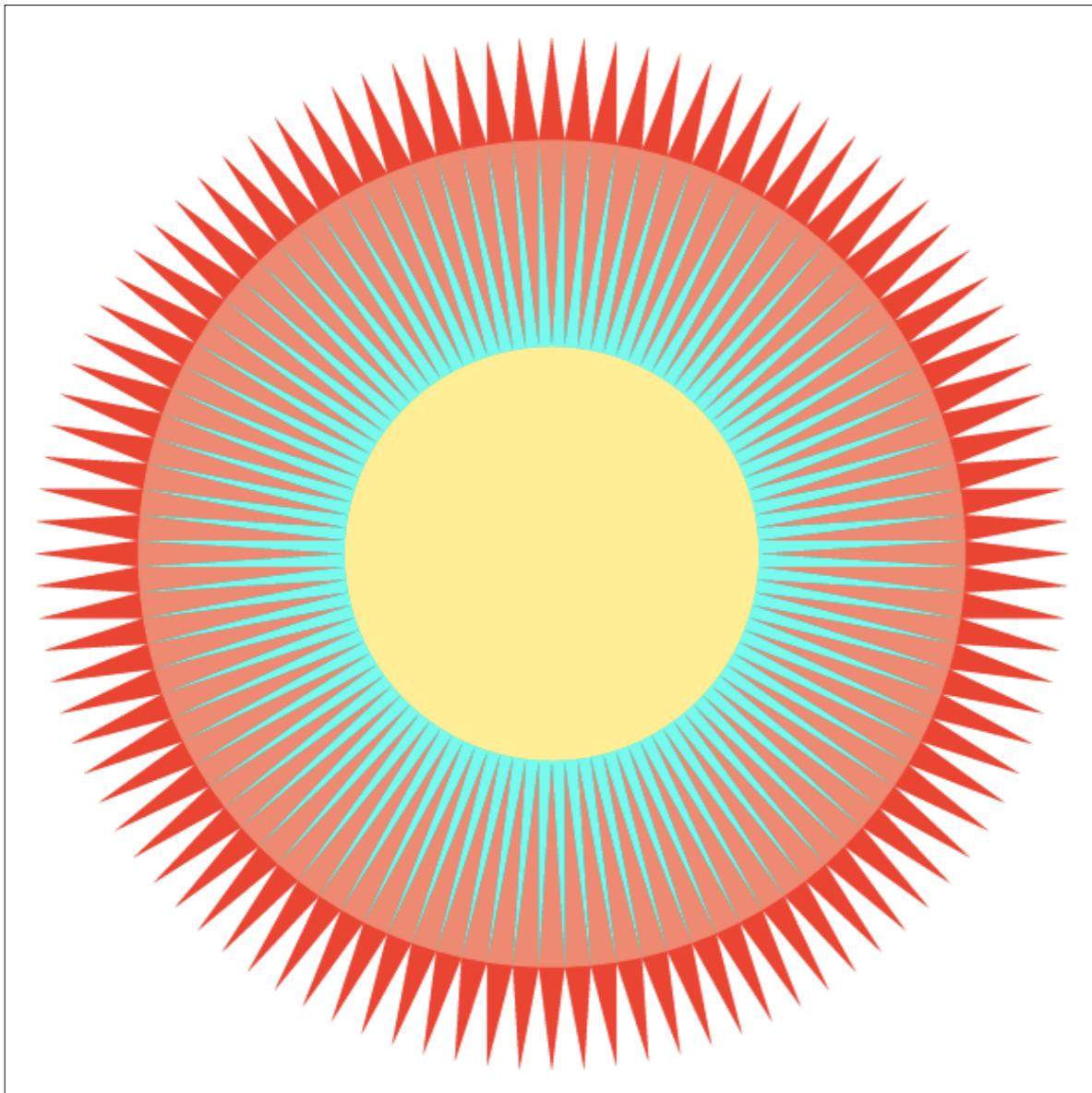
Sample

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

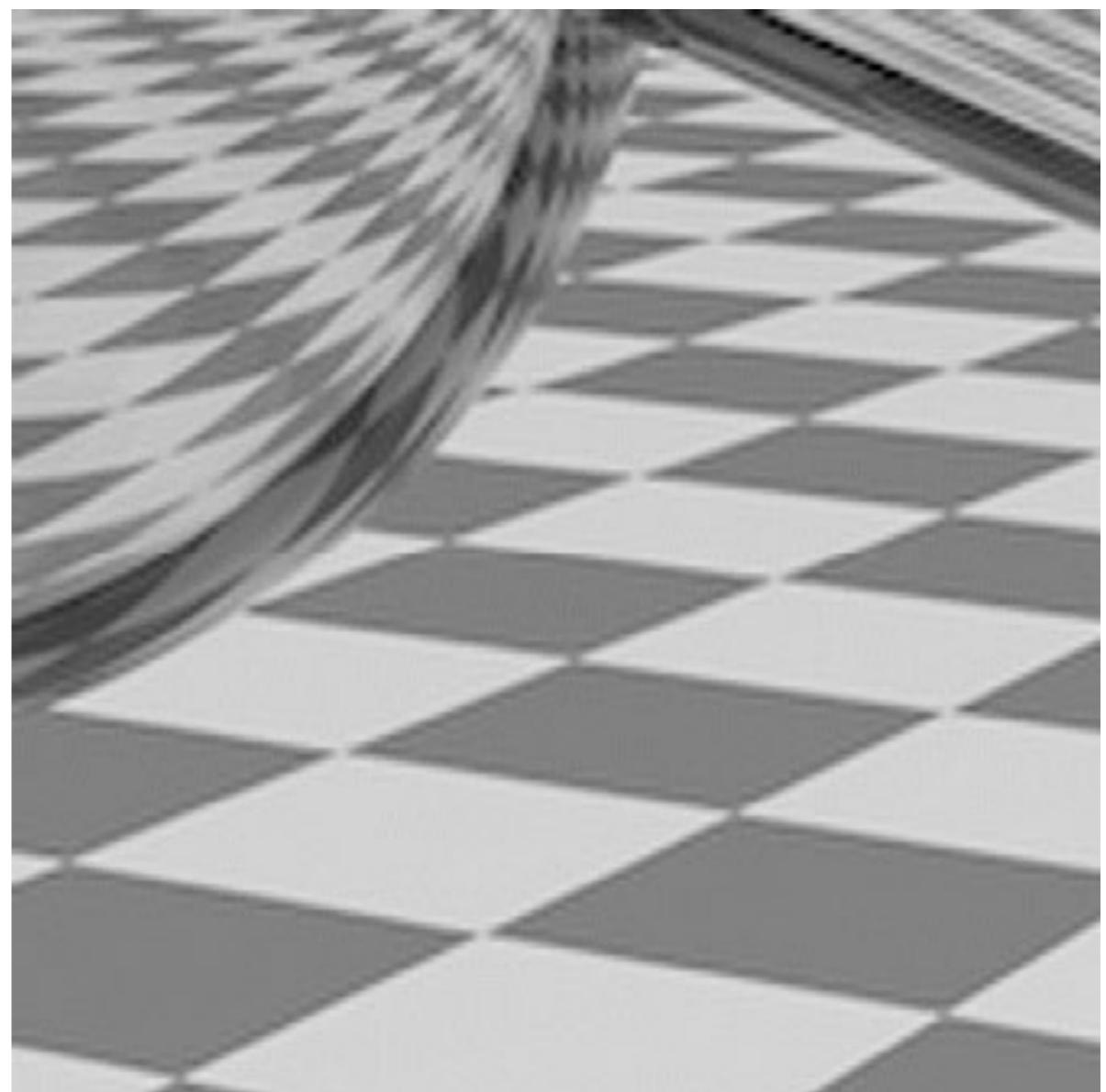
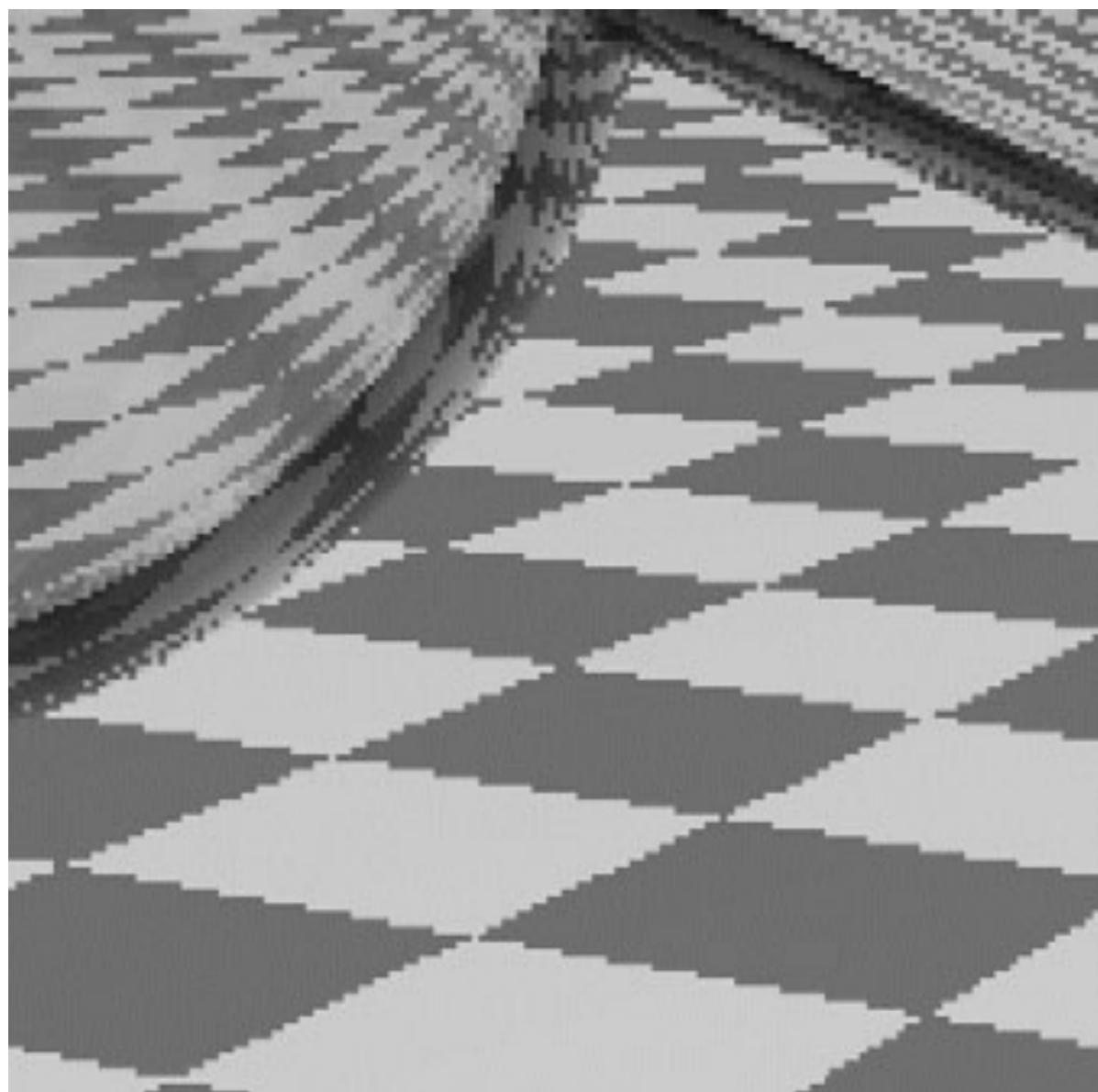
# Point Sampling



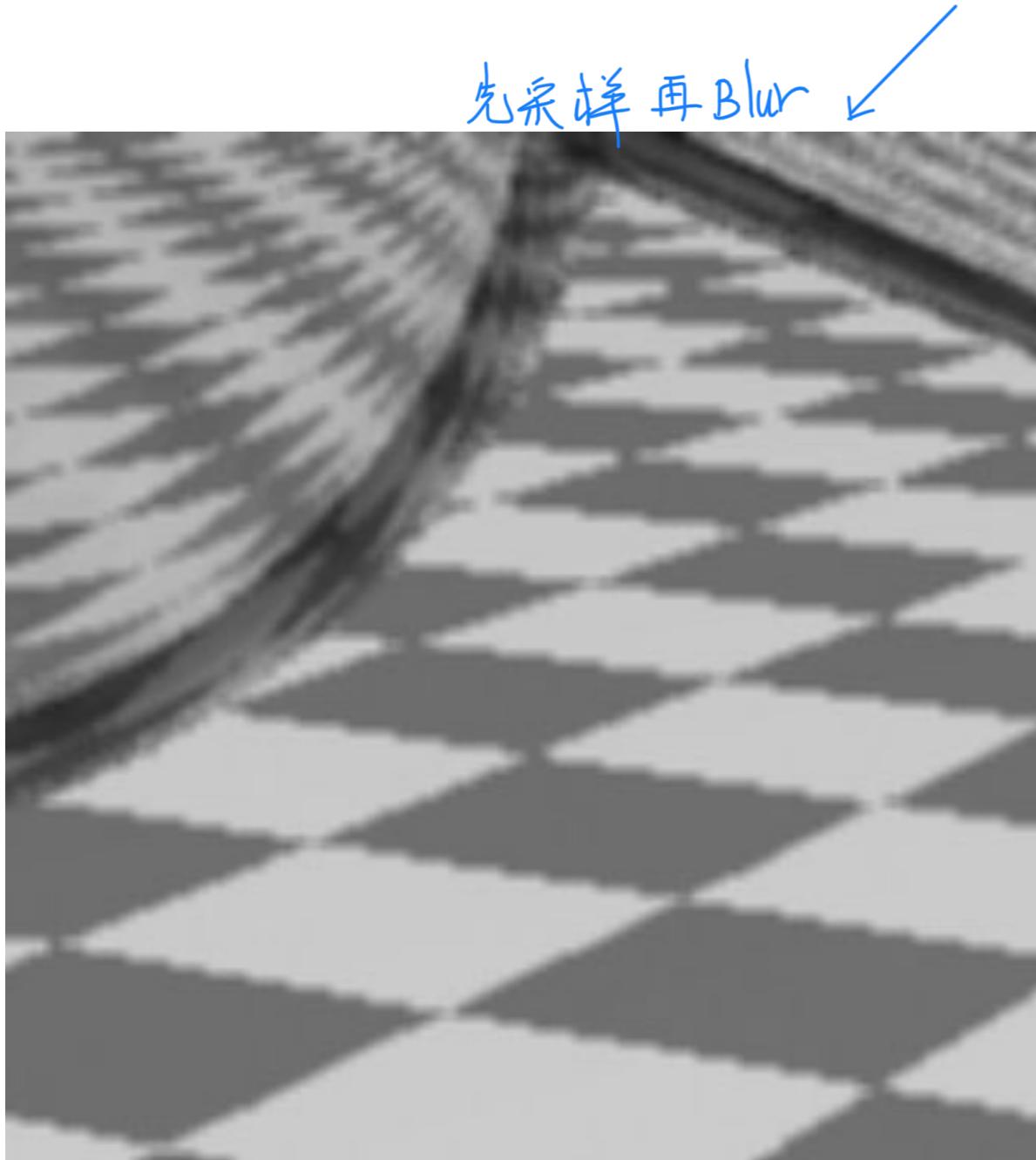
# Antialiasing



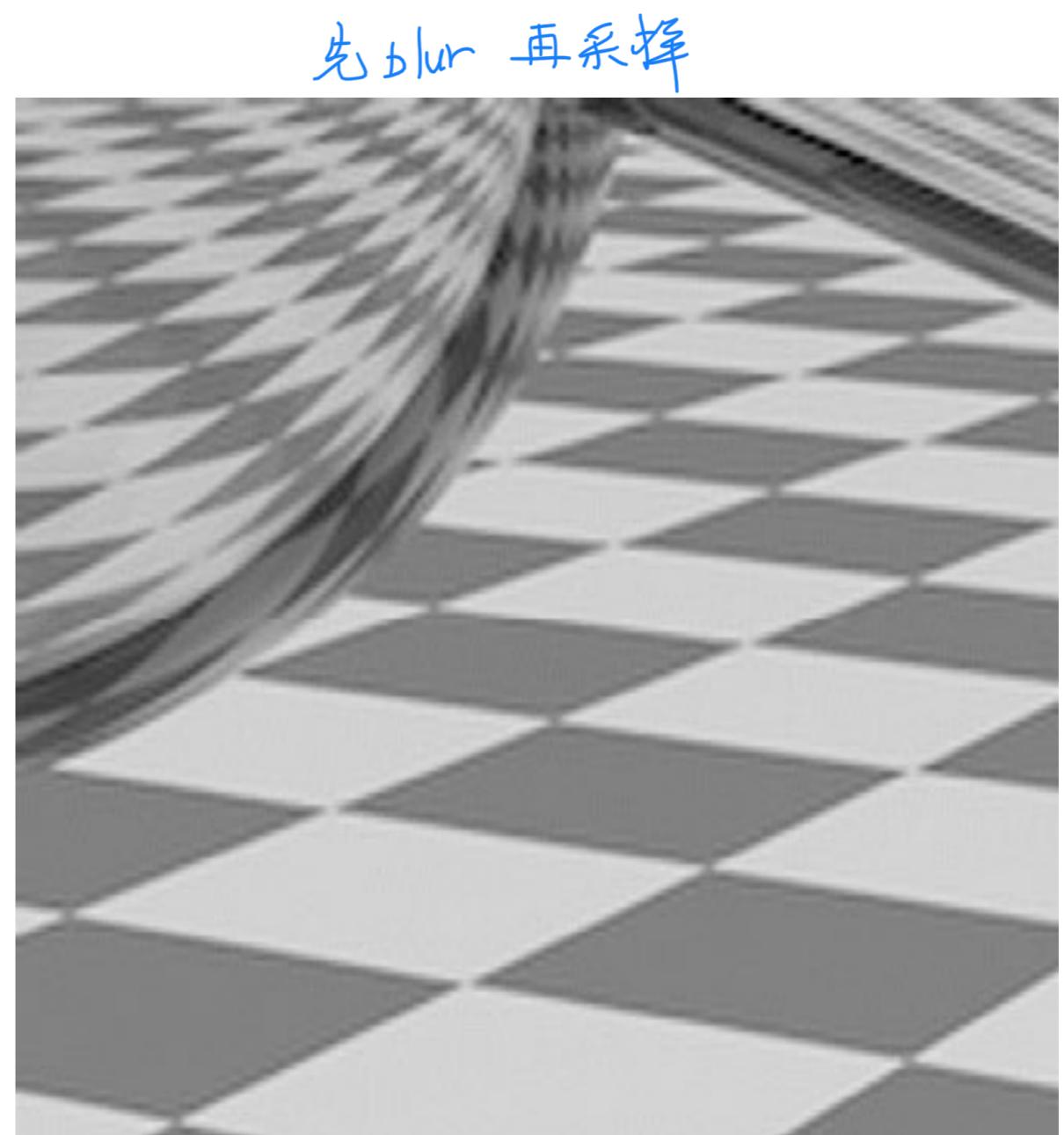
# Point Sampling vs Antialiasing



# 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?

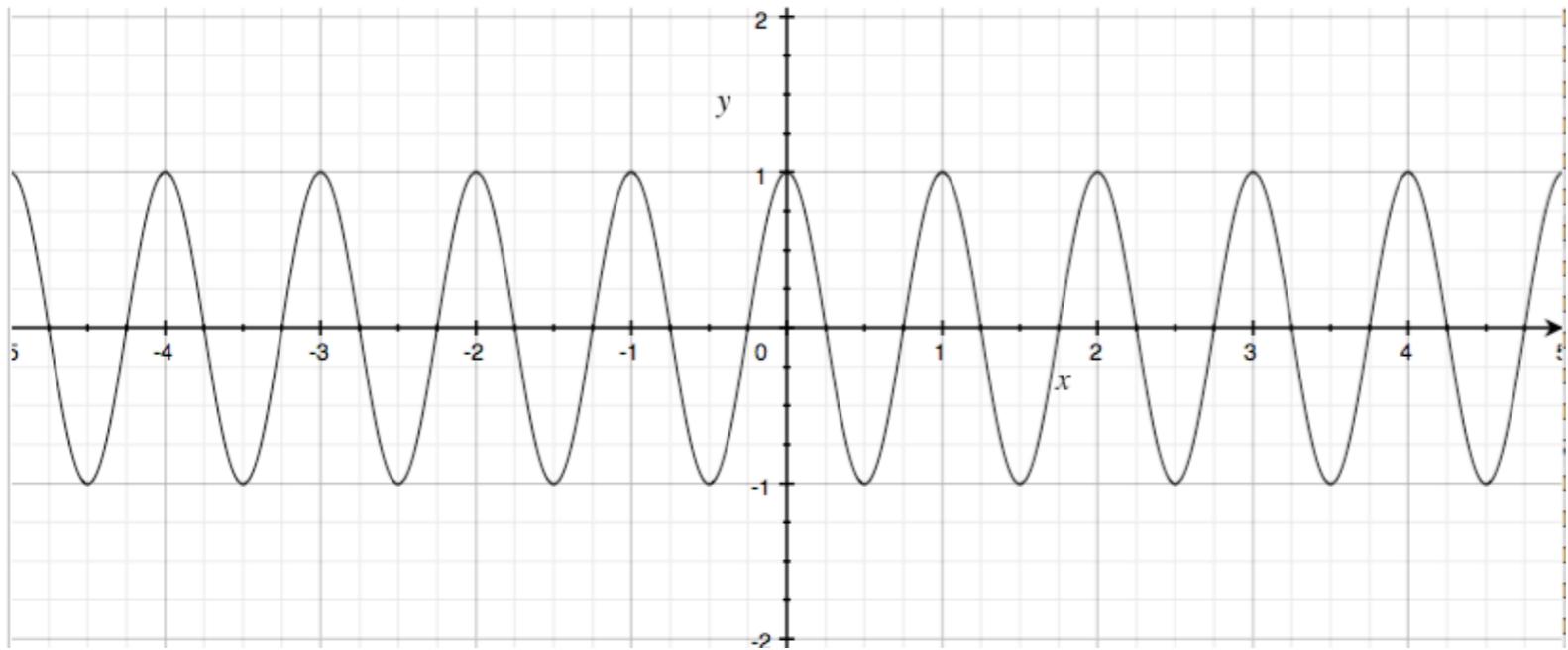
Let's dig into fundamental reasons

And look at how to implement antialiased rasterization

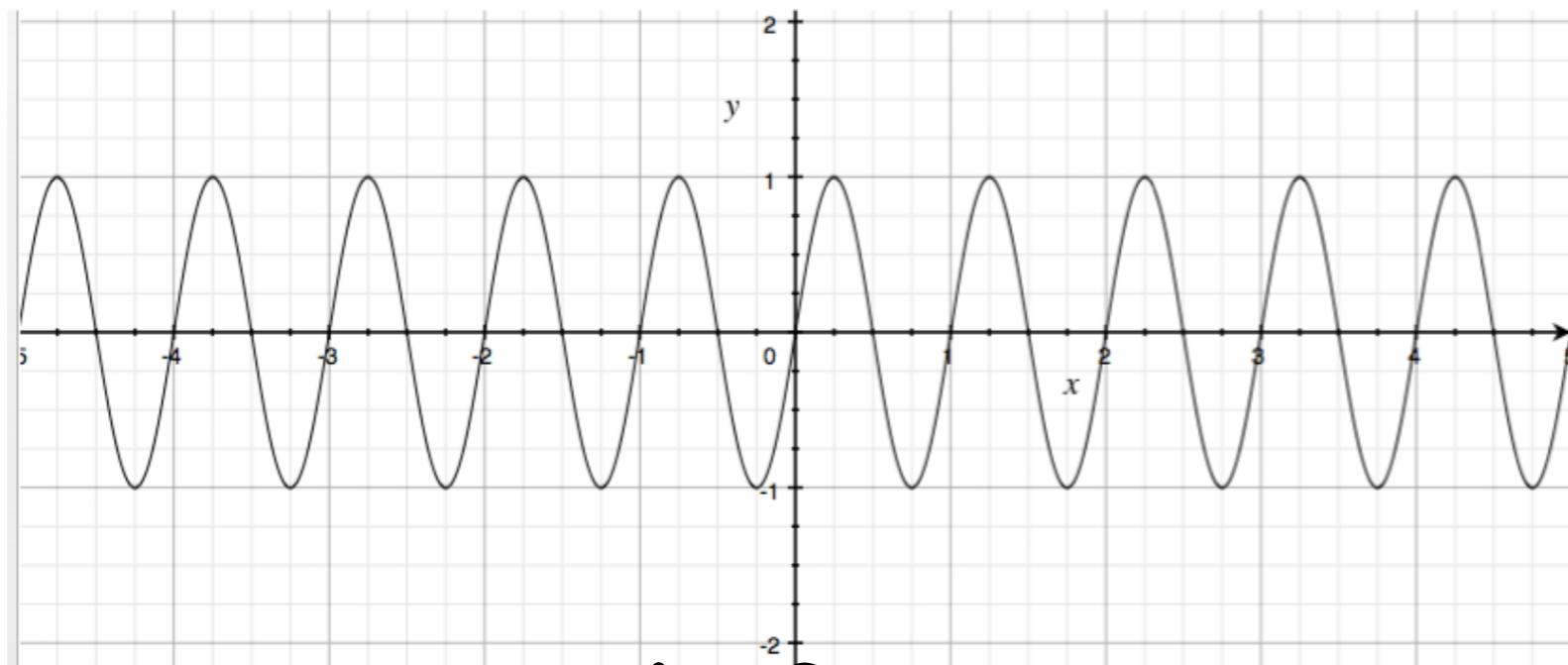
频域

# Frequency Domain

# Sines and Cosines



$$\cos 2\pi x$$

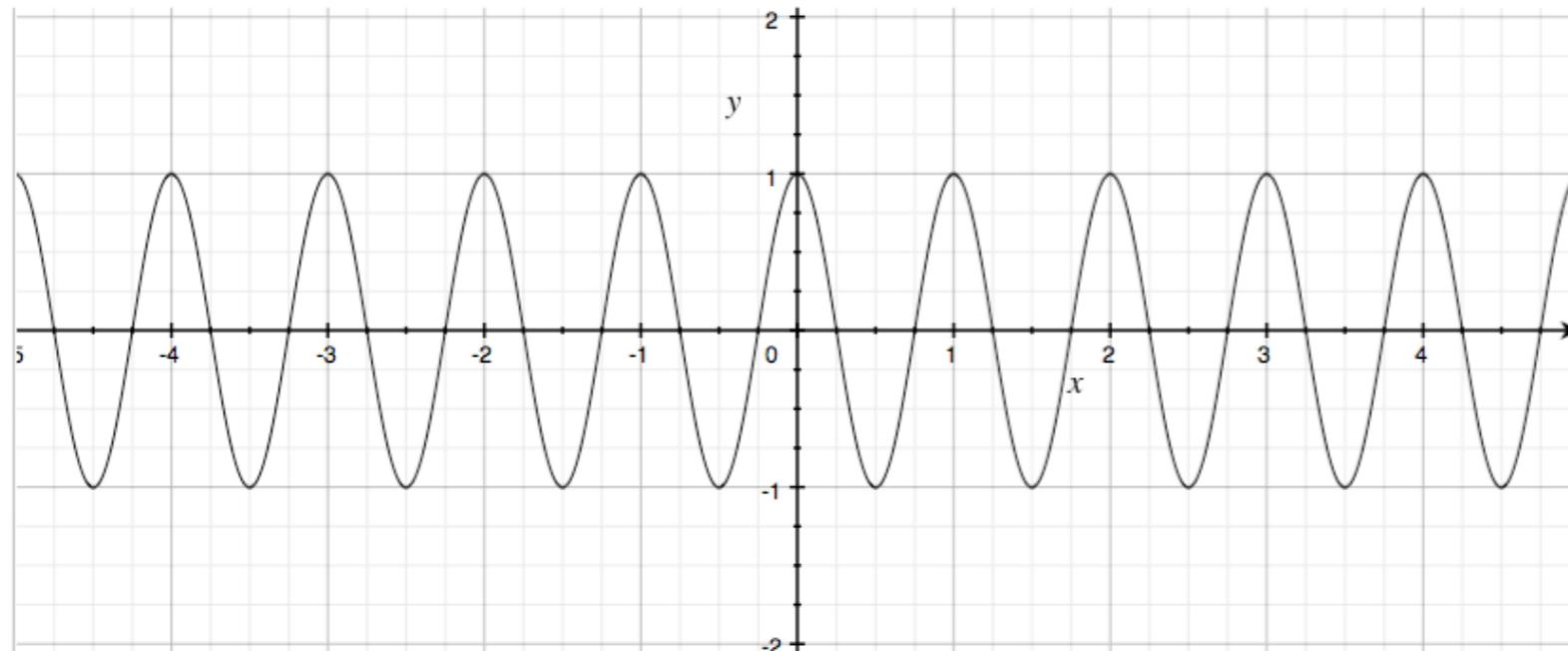


$$\sin 2\pi x$$

# Frequencies $\cos 2\pi f x$

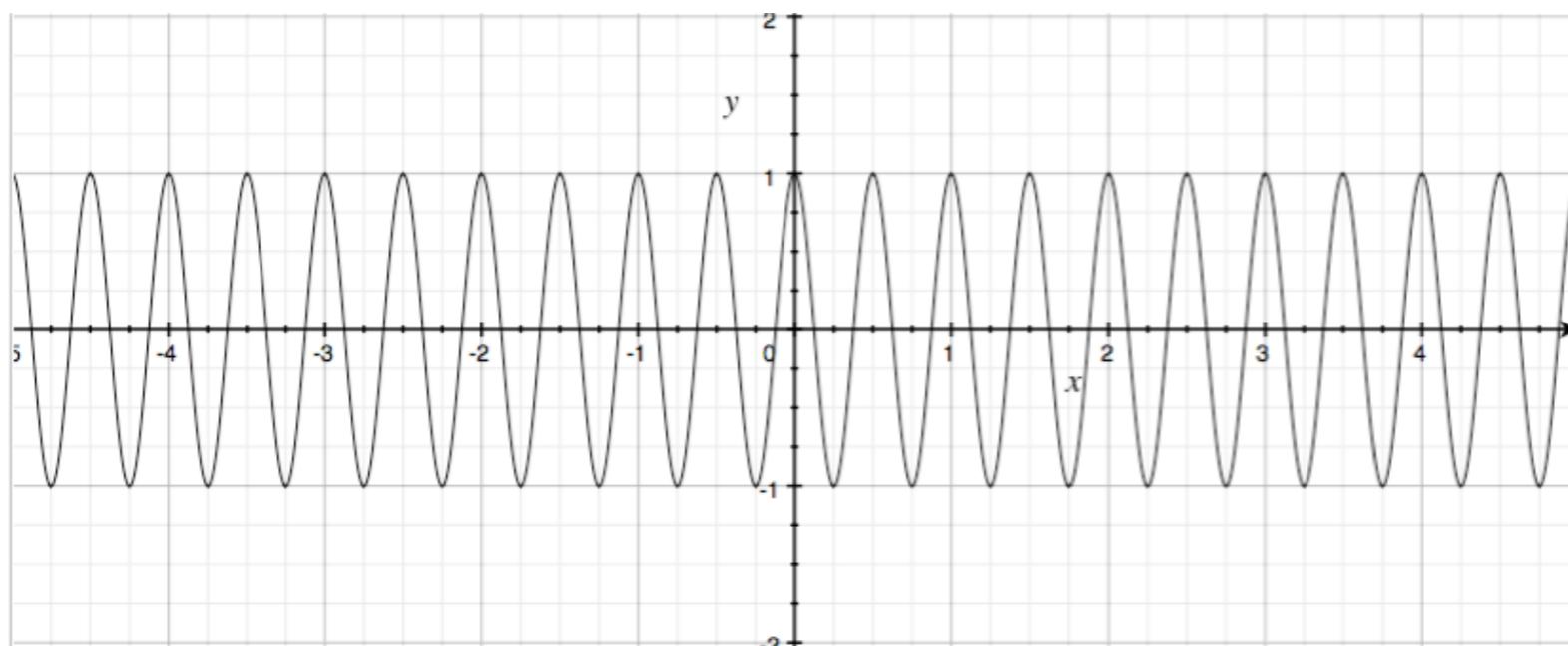
$$f = \frac{1}{T}$$

频率 周期



$$\cos 2\pi x$$

$$f = 1$$



$$\cos 4\pi x$$

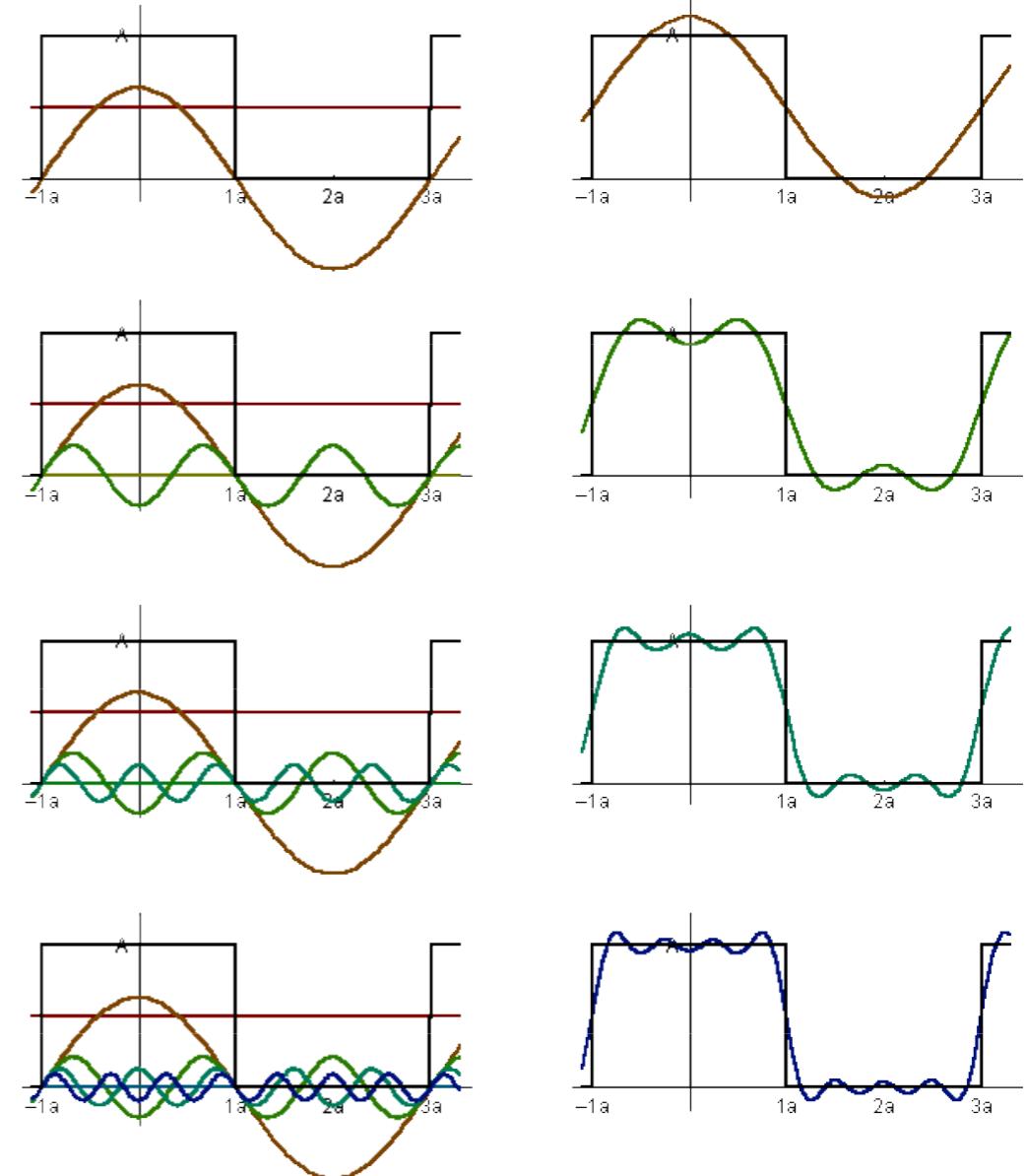
$$f = 2$$

# Fourier Transform

**Represent a function as a weighted sum of sines and cosines**



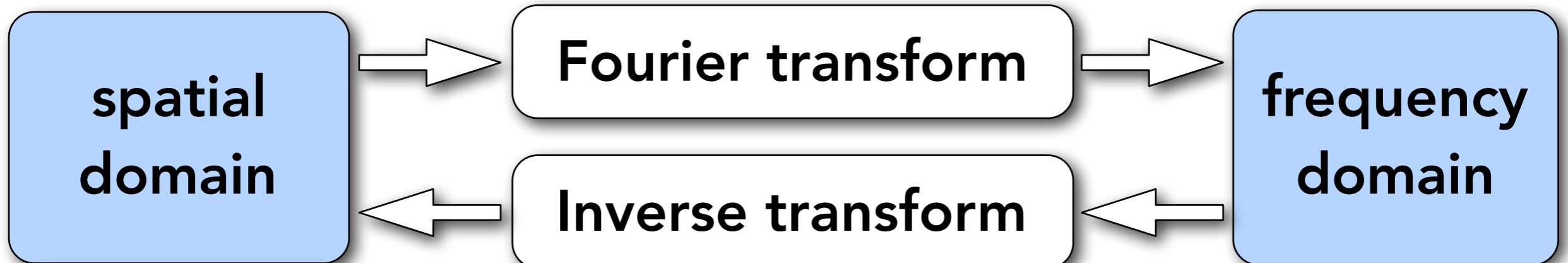
Joseph Fourier 1768 - 1830



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

# Fourier Transform Decomposes A Signal Into Frequencies

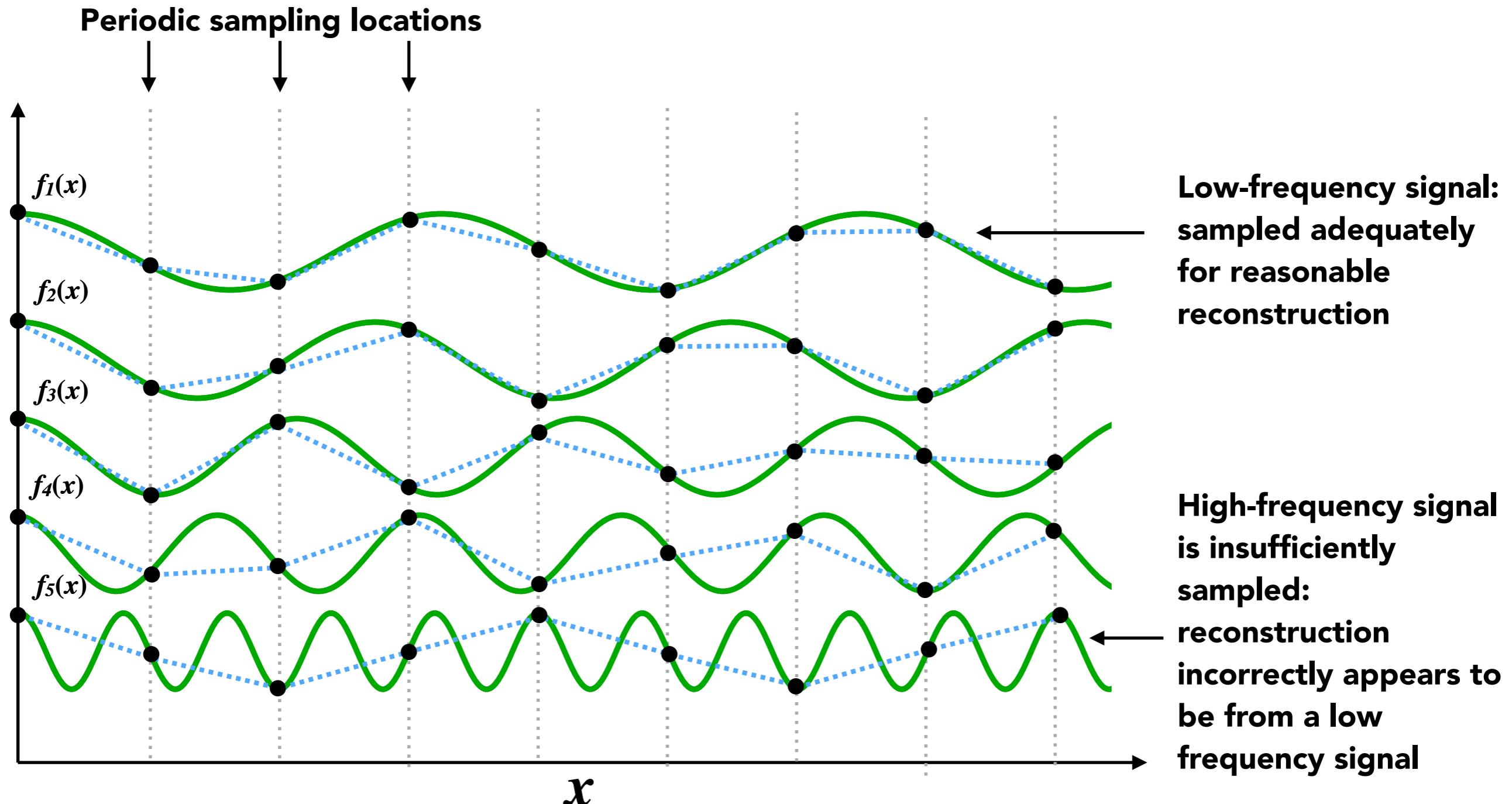
$$f(x) \quad F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i \omega x} dx \quad F(\omega)$$



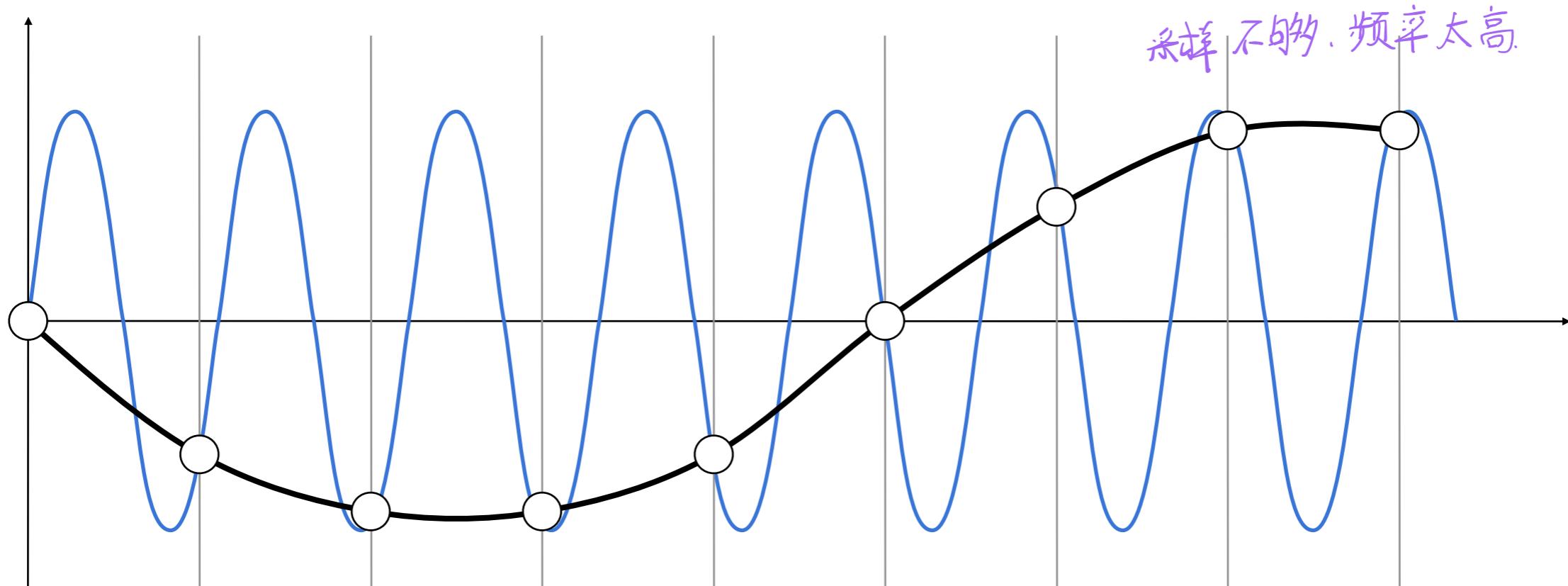
$$f(x) = \int_{-\infty}^{\infty} F(\omega) e^{2\pi i \omega x} d\omega$$

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

# Higher Frequencies Need Faster Sampling



# 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”

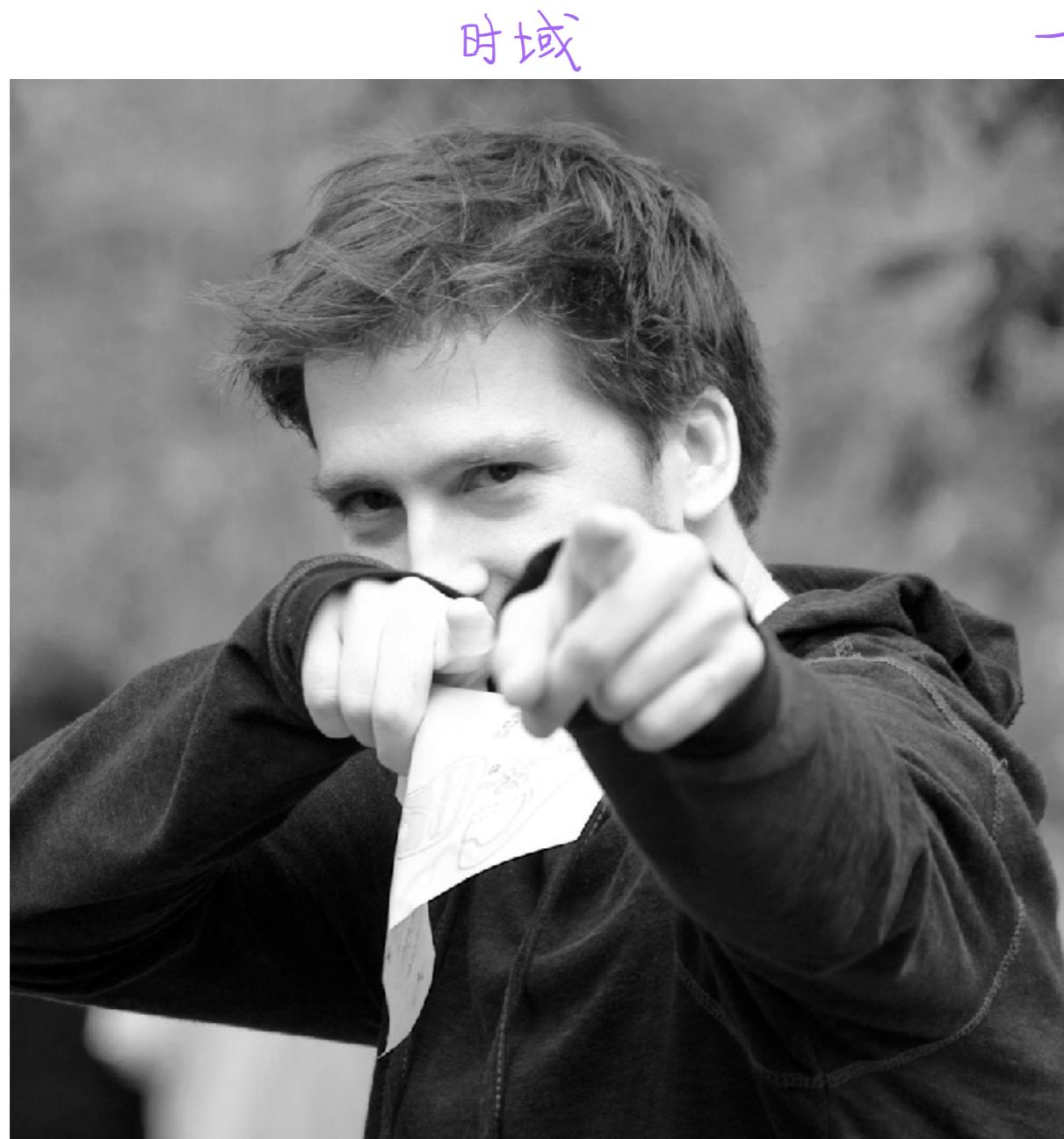
同样采样方法采样两种不同函数，我们无法区分（走样）

滤波

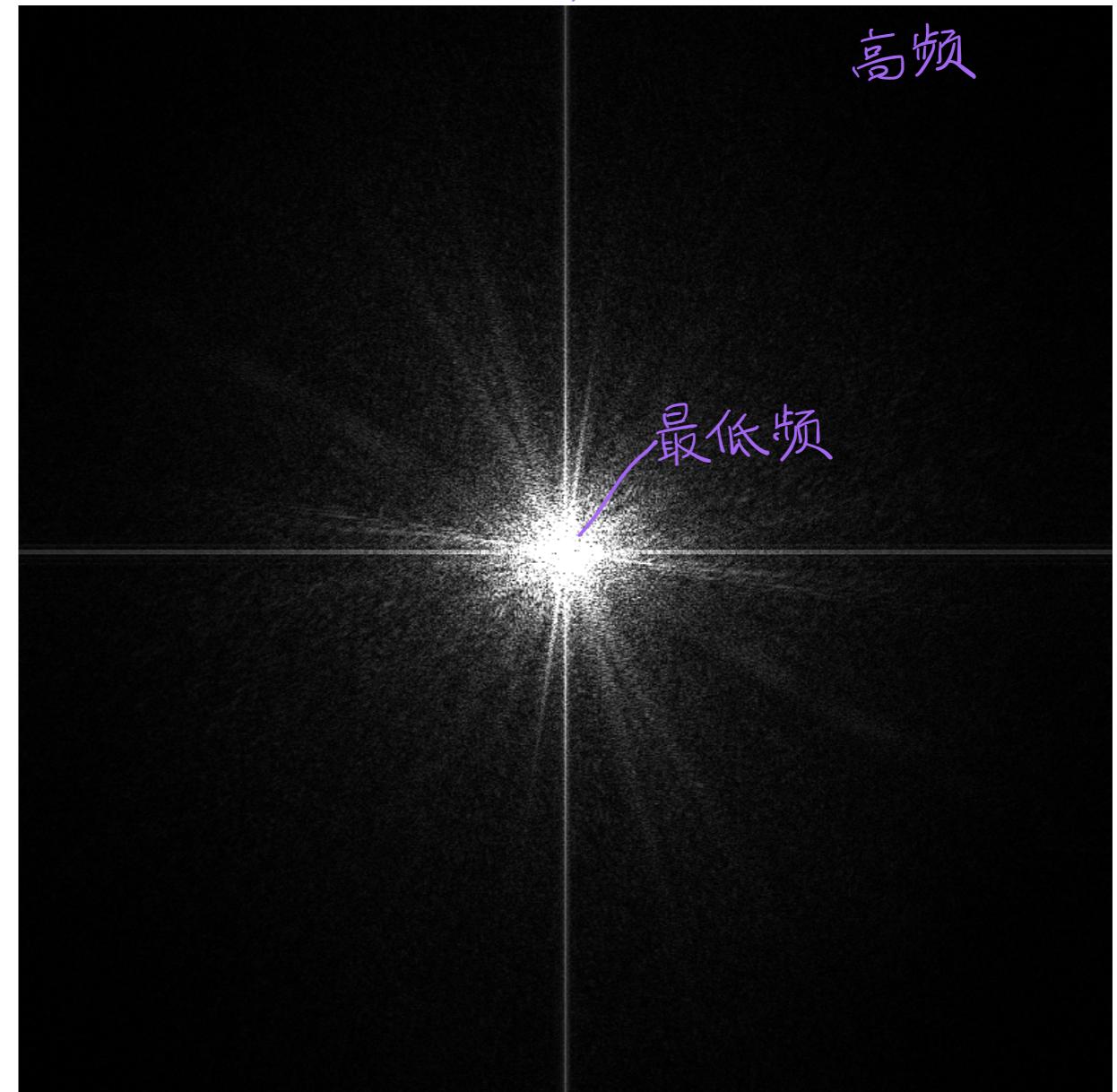
删掉特定频率的信号

Filtering = Getting rid of  
certain frequency contents

# Visualizing Image Frequency Content

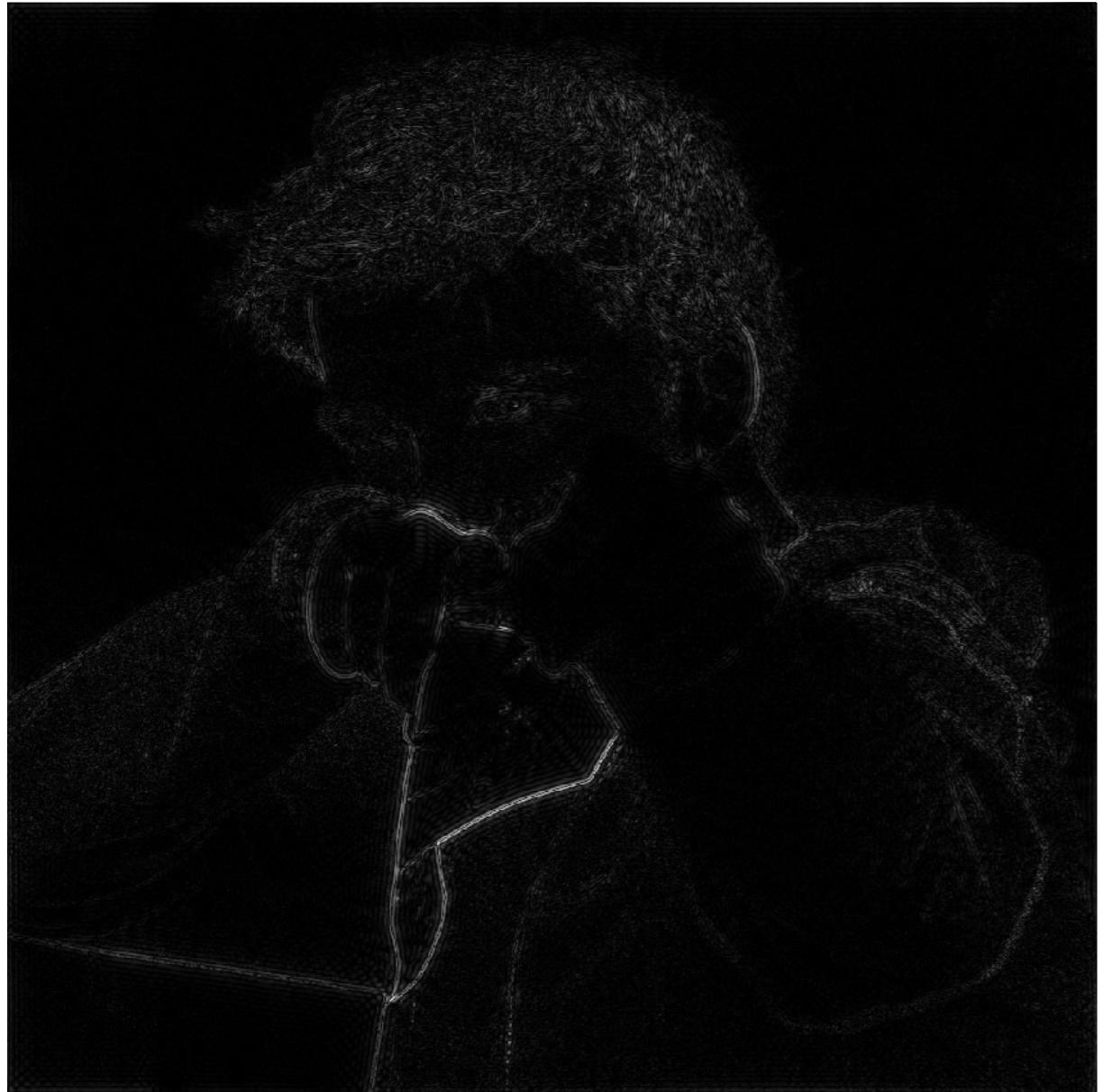


→ 频域

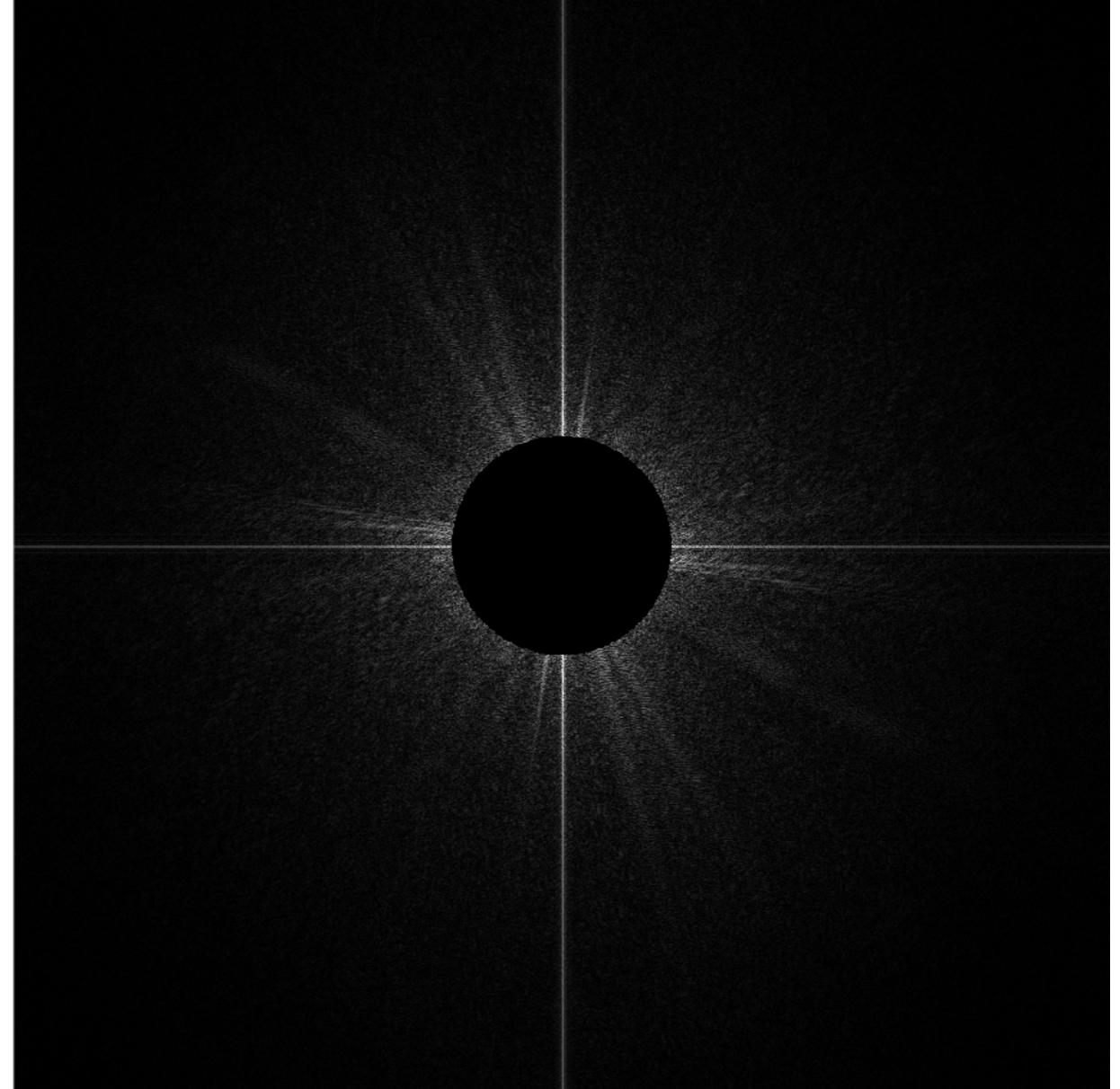


亮度高 信息多  
一般图片低频信息多  
高频信息少

# Filter Out Low Frequencies Only (Edges)



**High-pass filter** (图像边界)

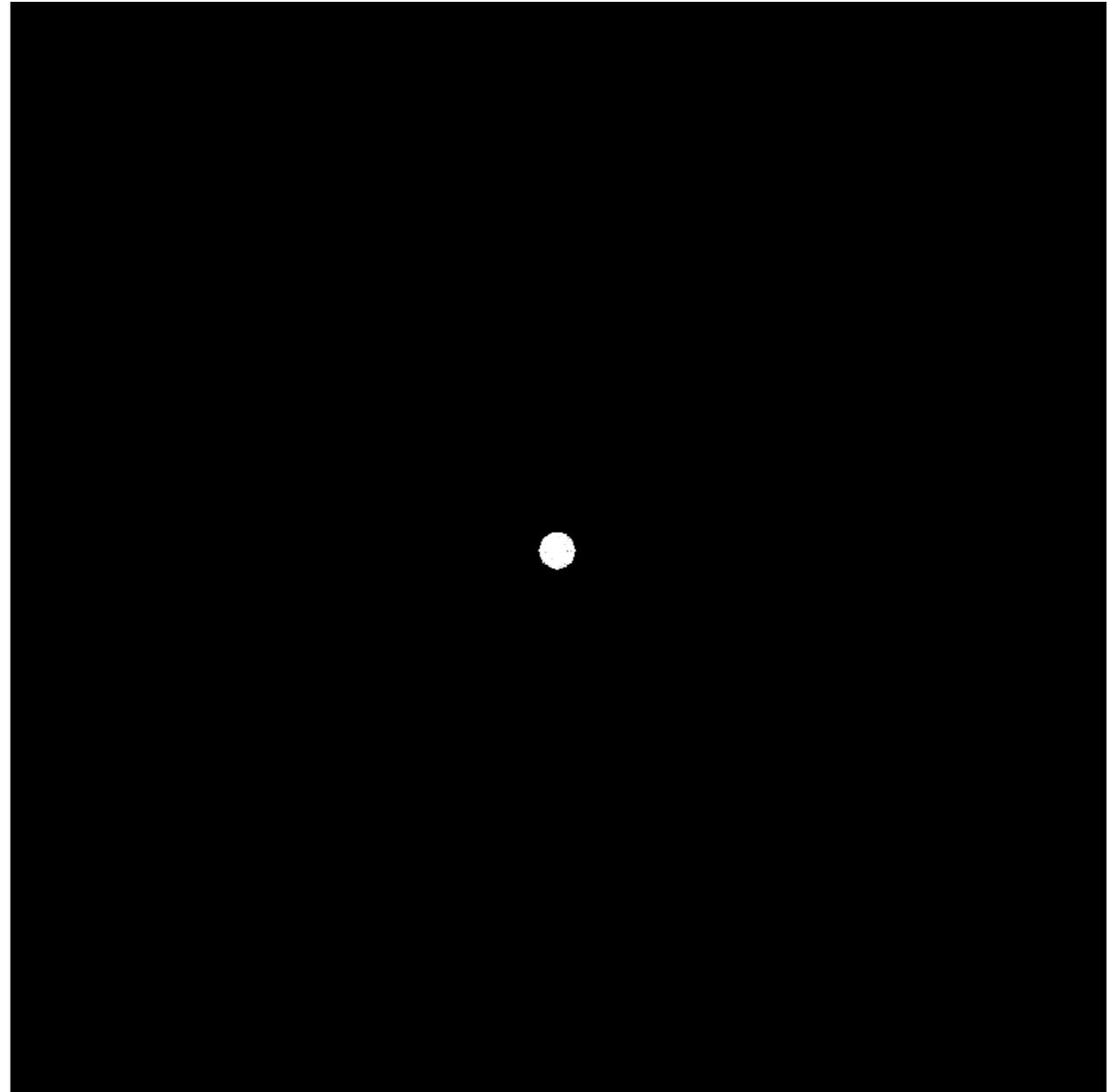


高通滤波

# Filter Out High Frequencies (Blur)



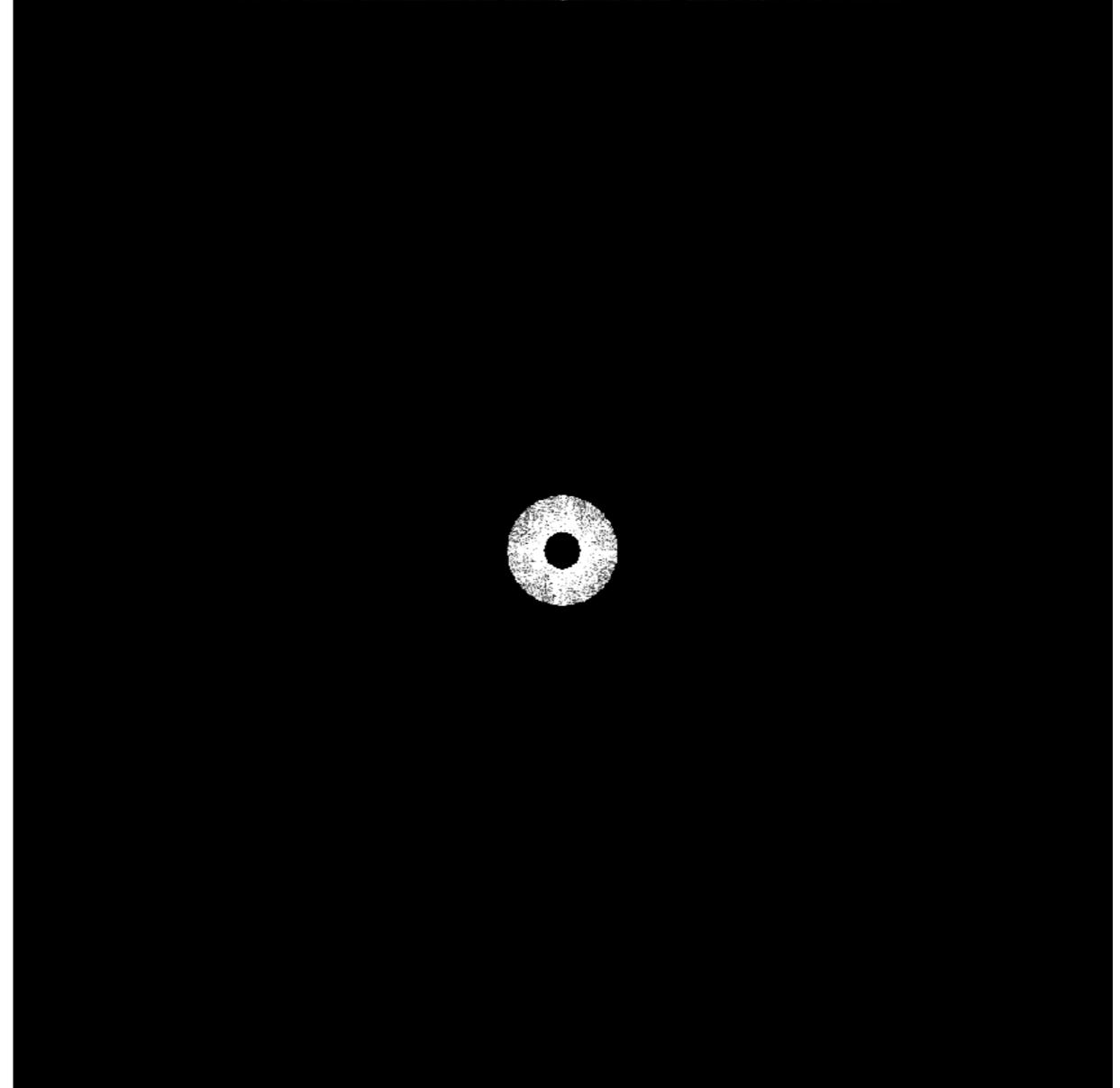
**Low-pass filter**



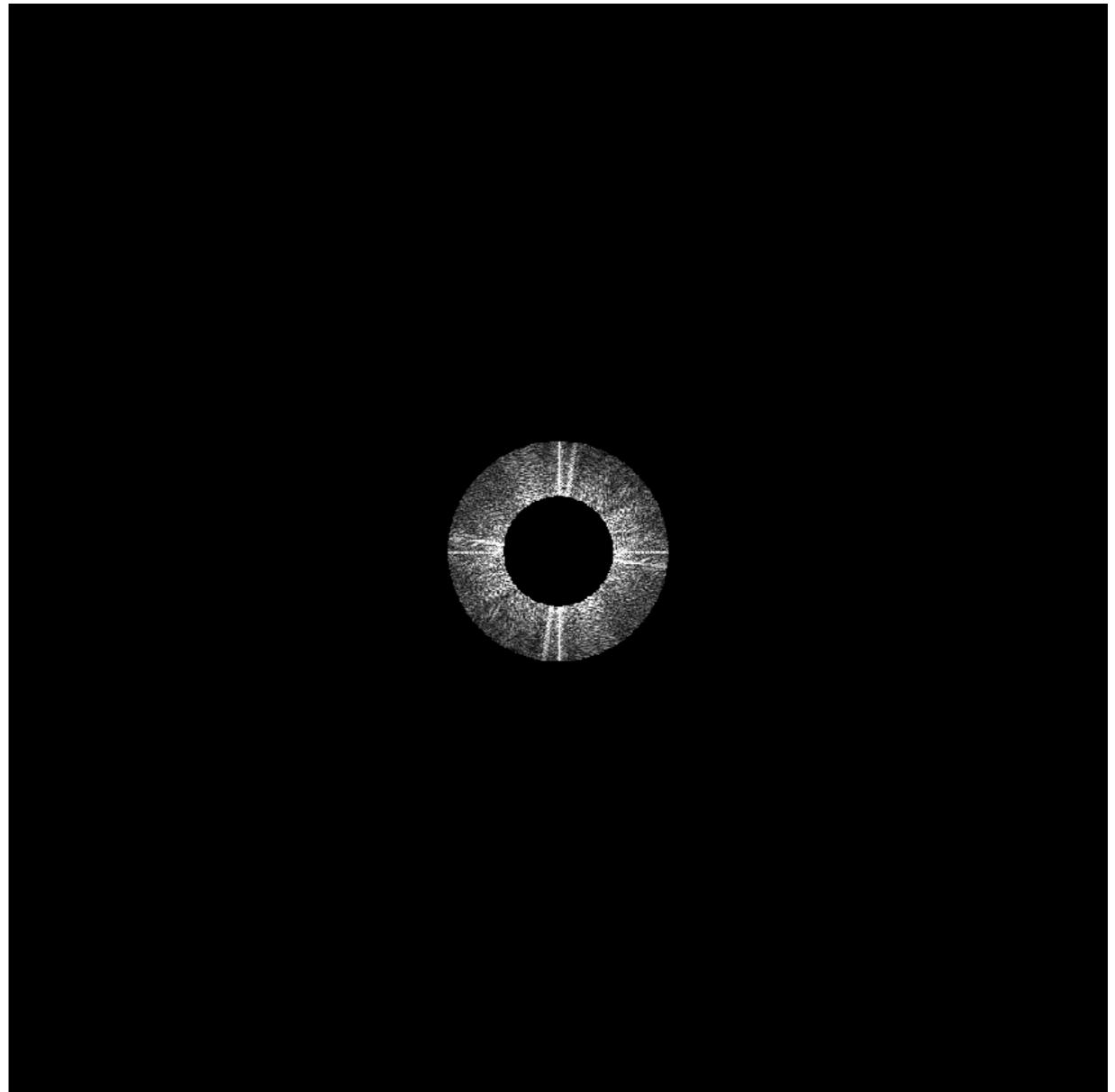
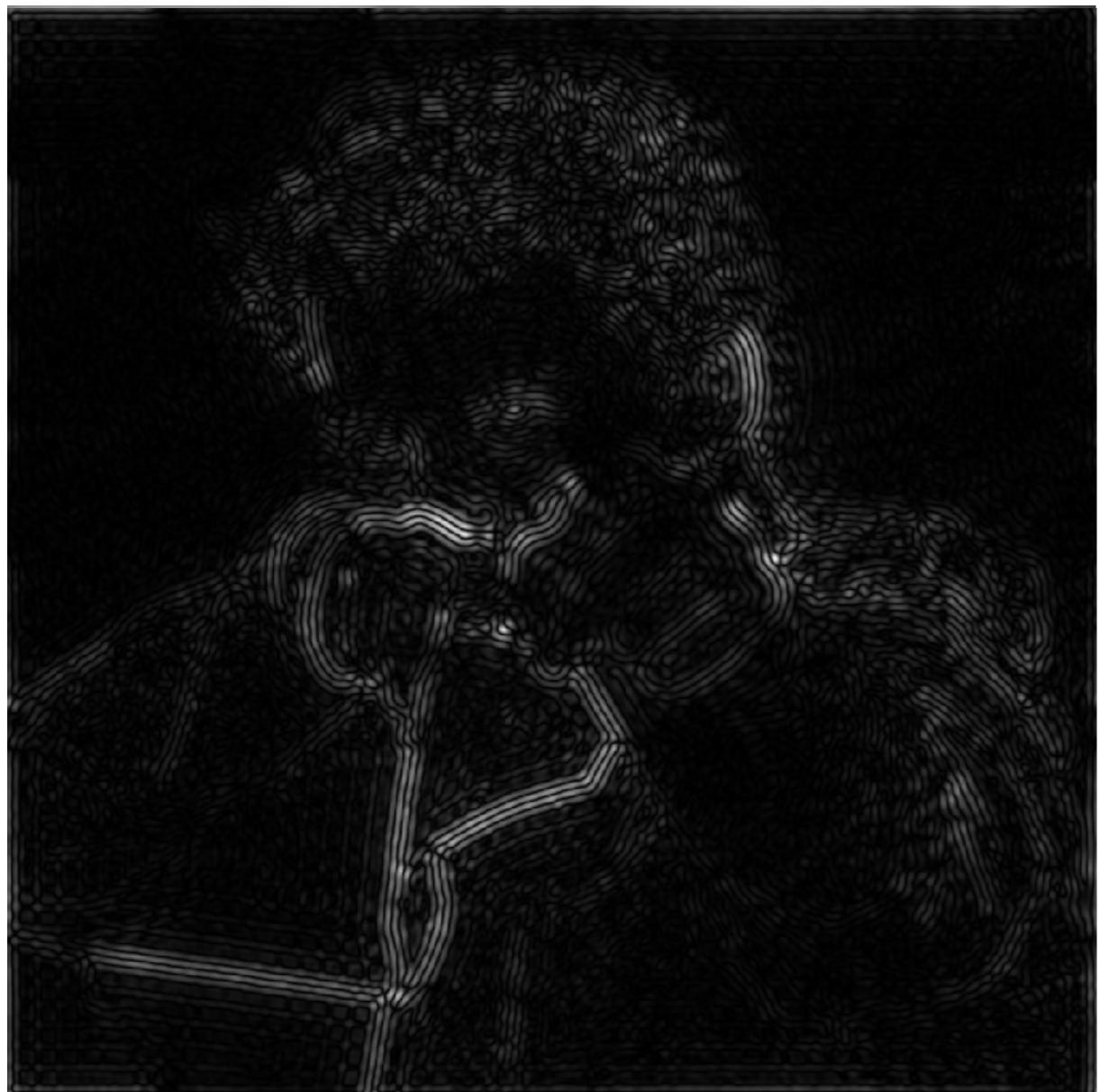
低通濾波

(只保留低頻  
信號)

# Filter Out Low and High Frequencies



# Filter Out Low and High Frequencies

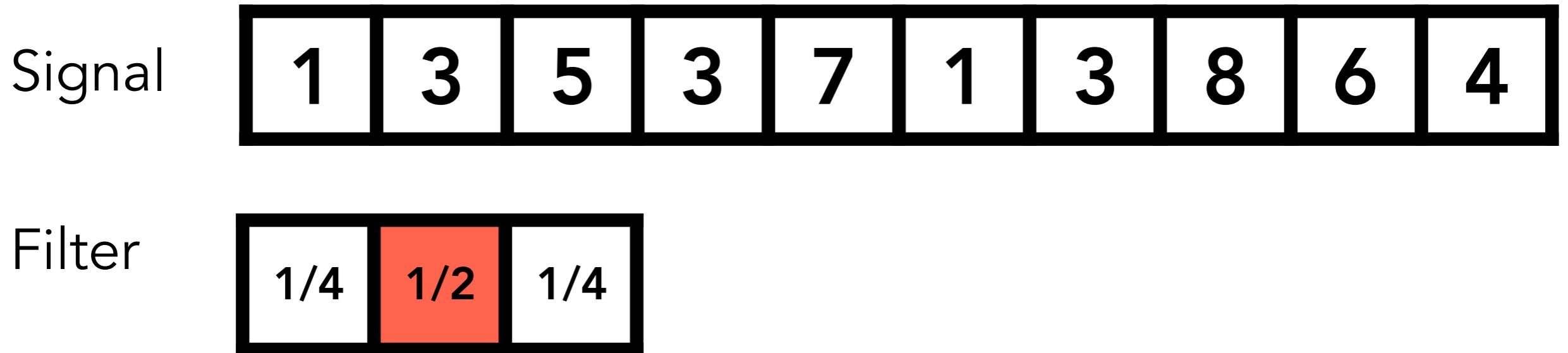


Filtering = Convolution  
( $=$  Averaging)

卷积

平均

# Convolution



Point-wise local averaging in a “sliding window”

# Convolution



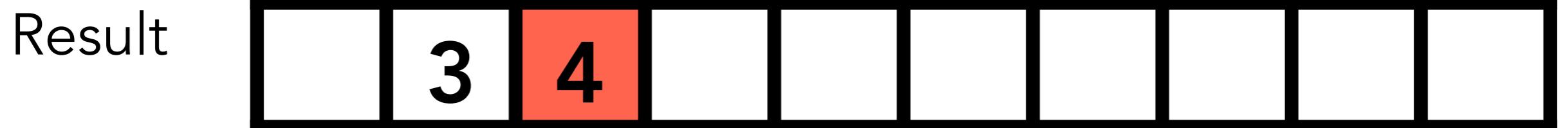
$$1 \times (1/4) + 3 \times (1/2) + 5 \times (1/4) = 3$$



# Convolution



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



# Convolution Theorem

时域上卷积  $\Leftrightarrow$  频域上乘积

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

Option 1:

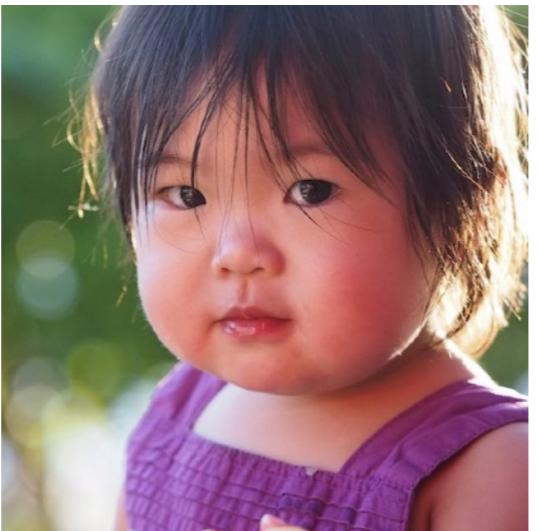
- Filter by convolution in the spatial domain

Option 2:

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

# Convolution Theorem

Spatial  
Domain



$$\ast \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} =$$

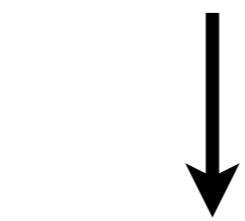


Fourier  
Transform

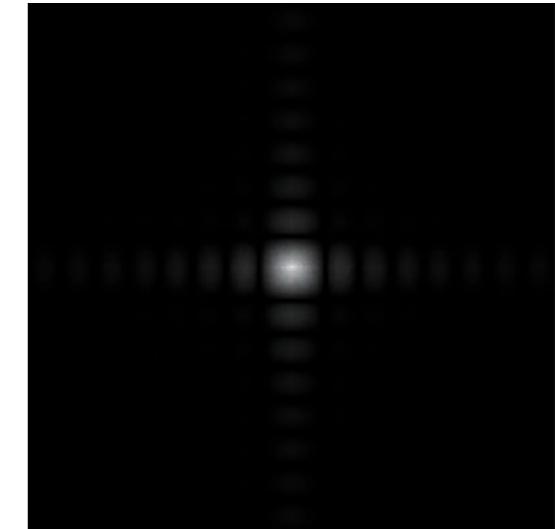
Frequency  
Domain



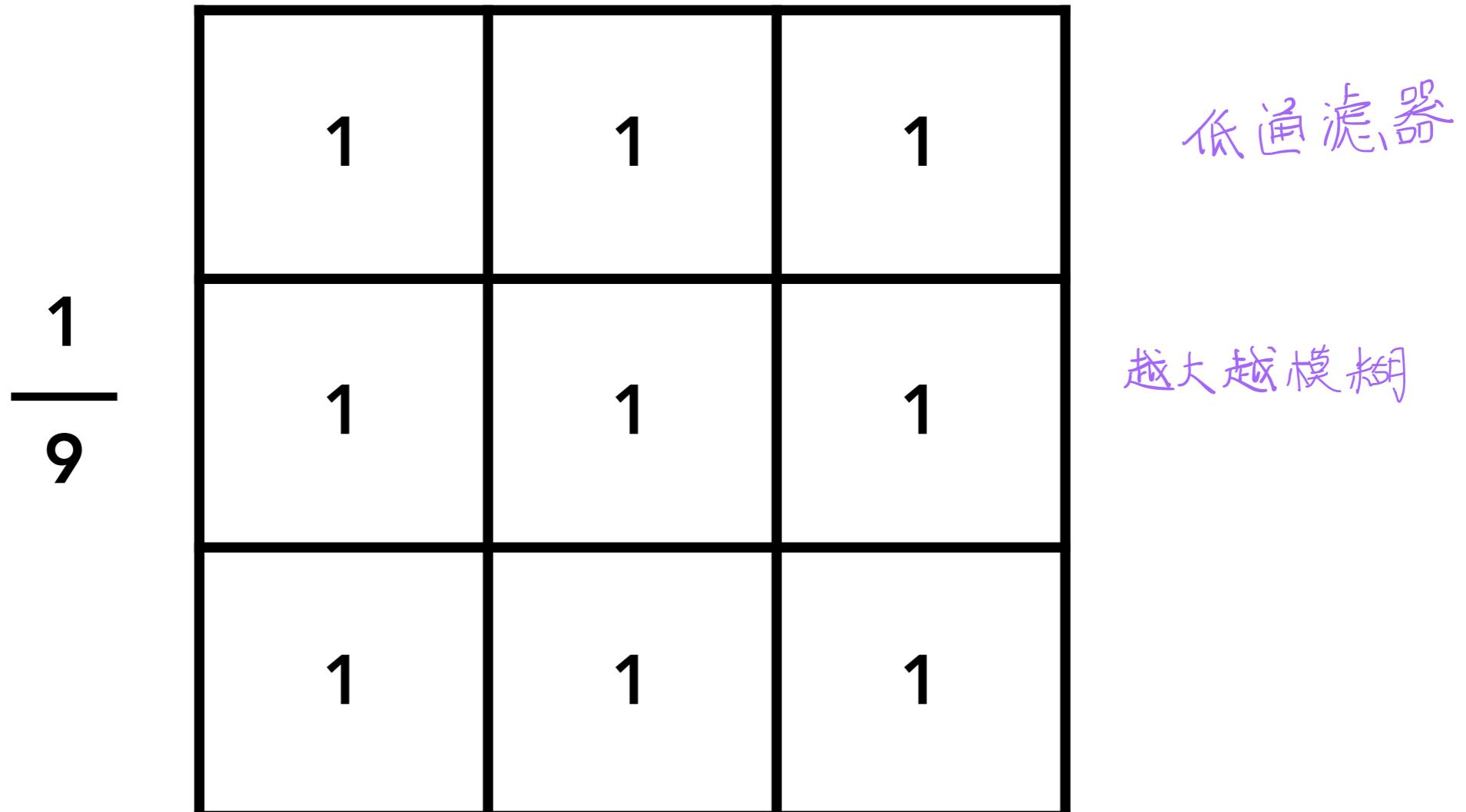
$$\times$$



Inv. Fourier  
Transform

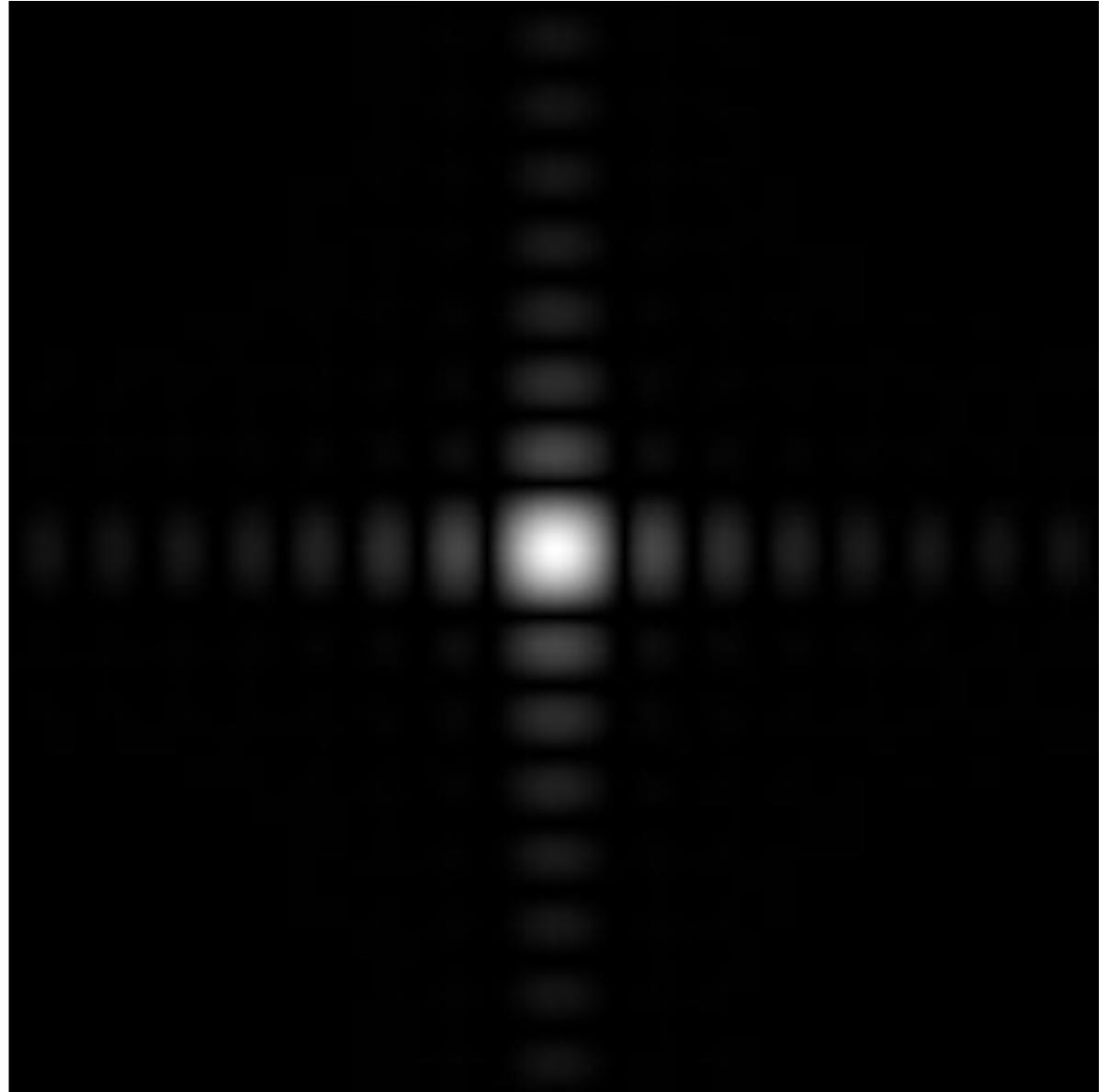
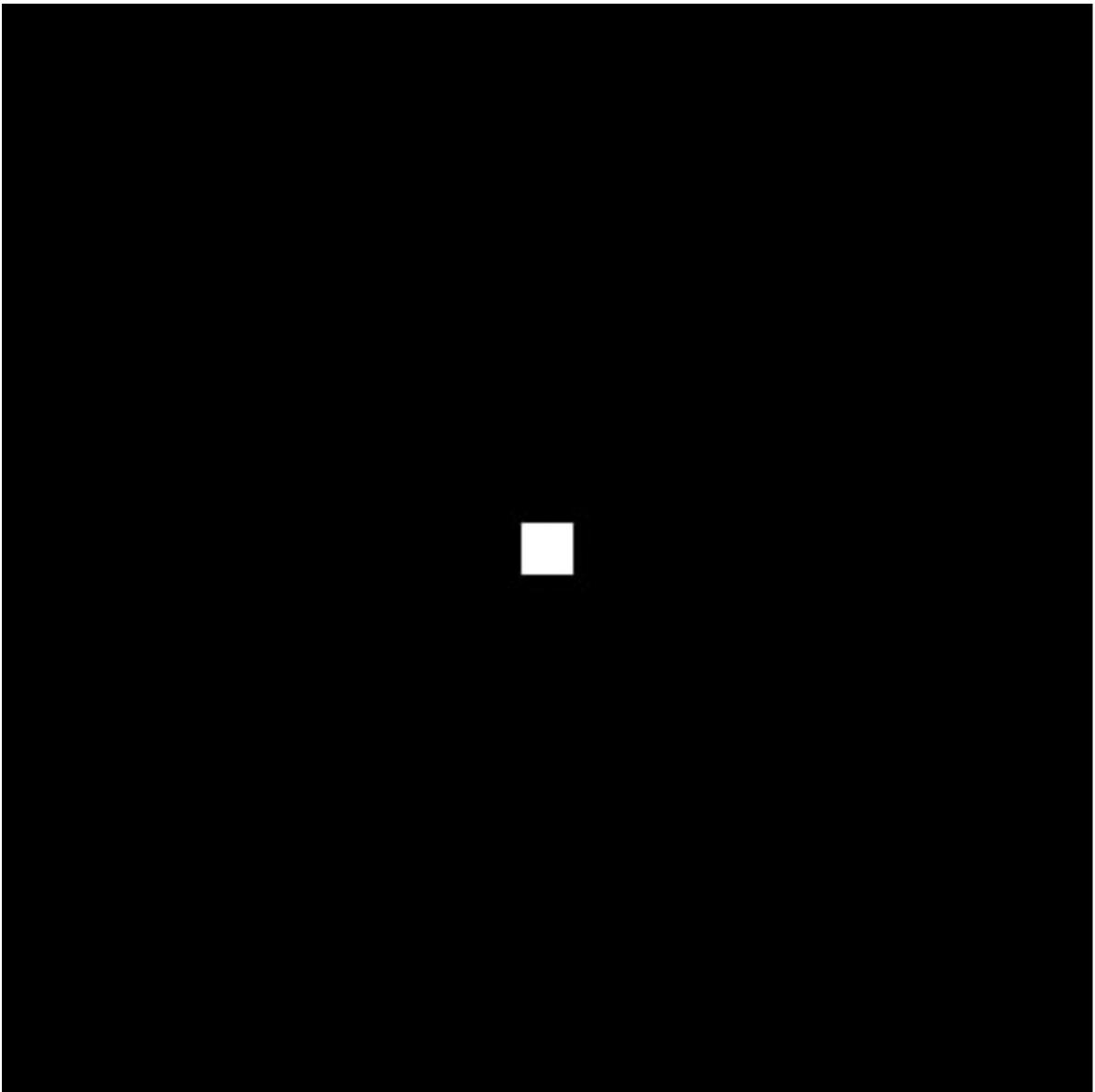


# Box Filter

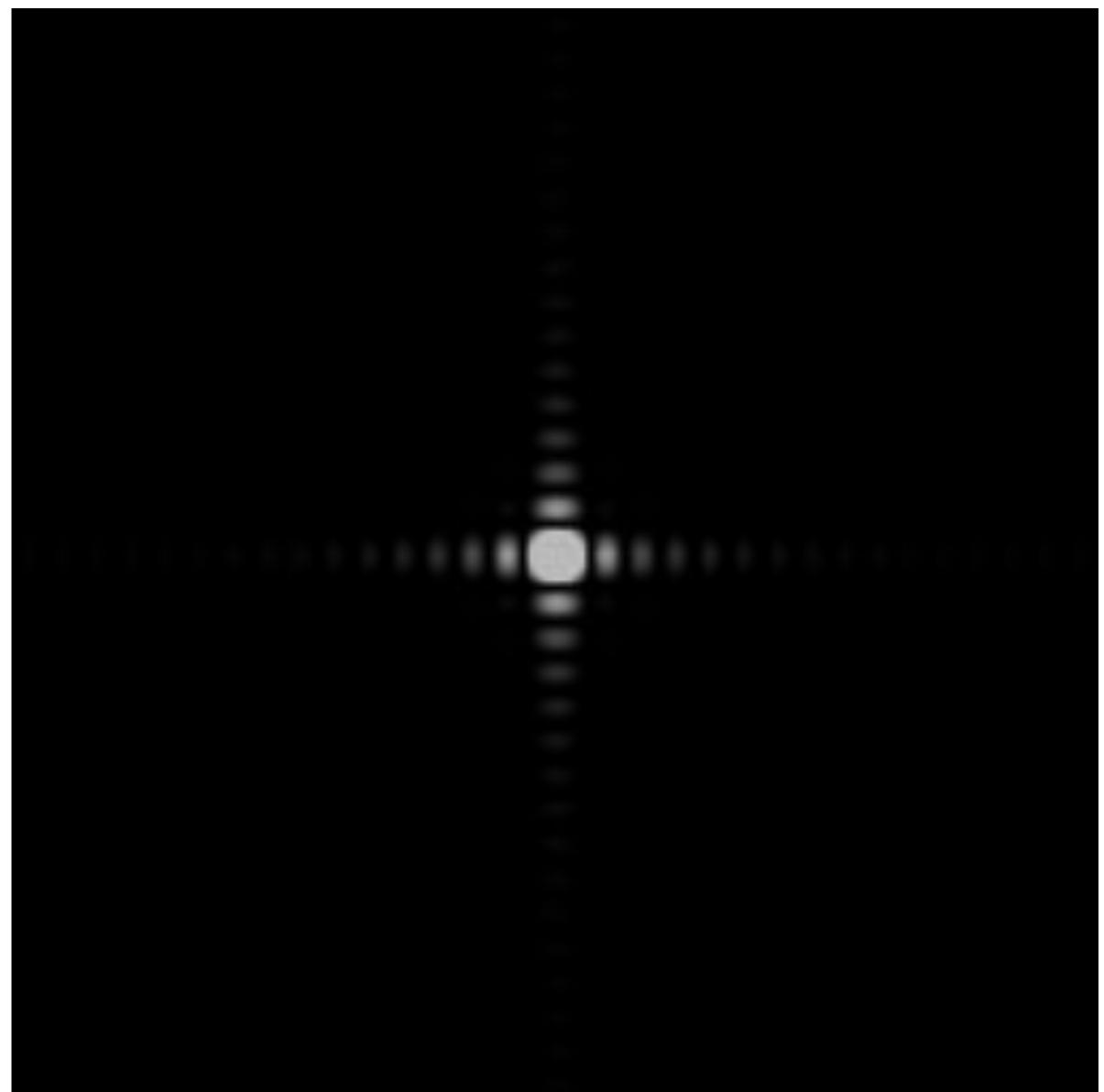
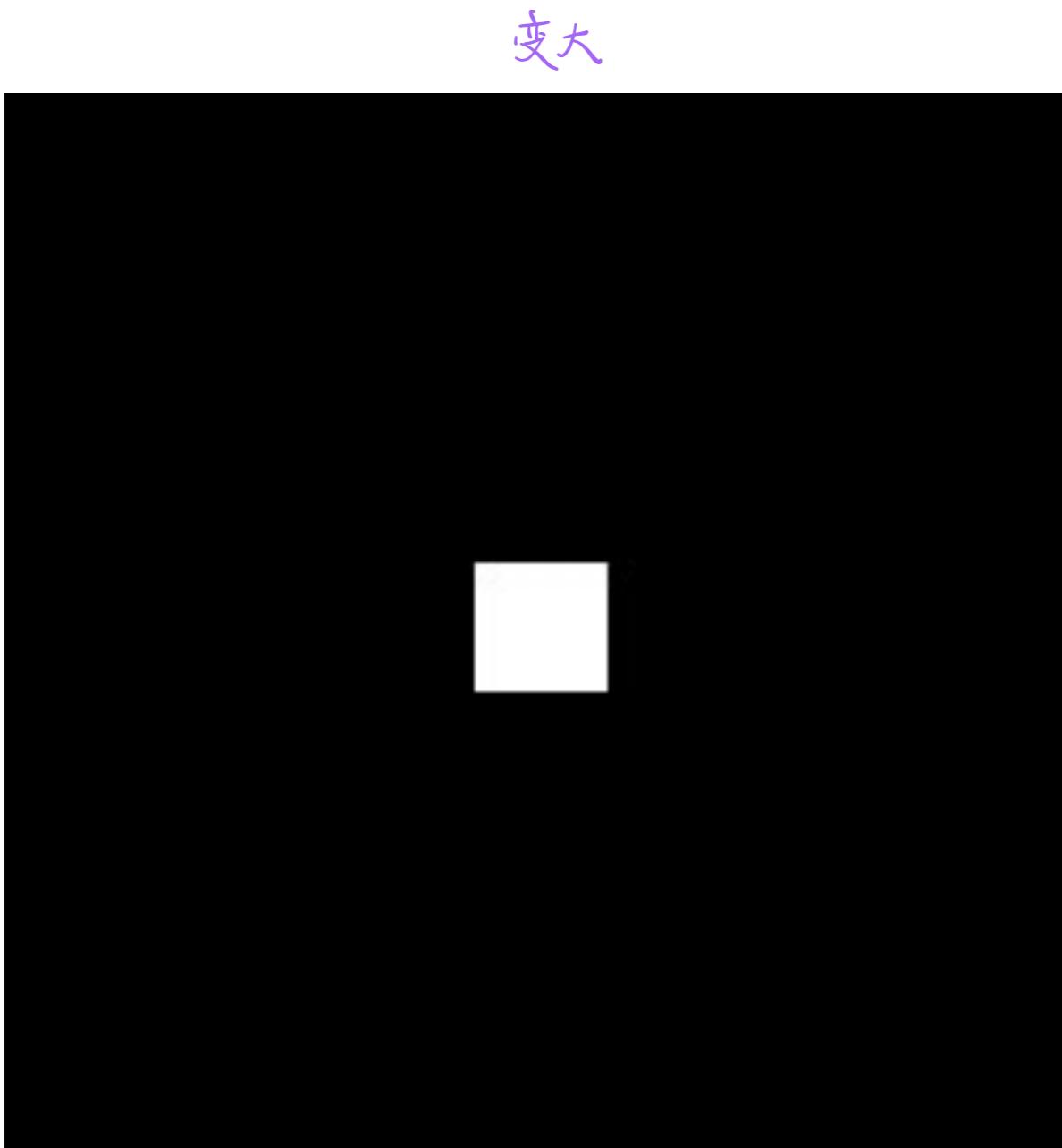


**Example: 3x3 box filter**

# Box Function = “Low Pass” Filter

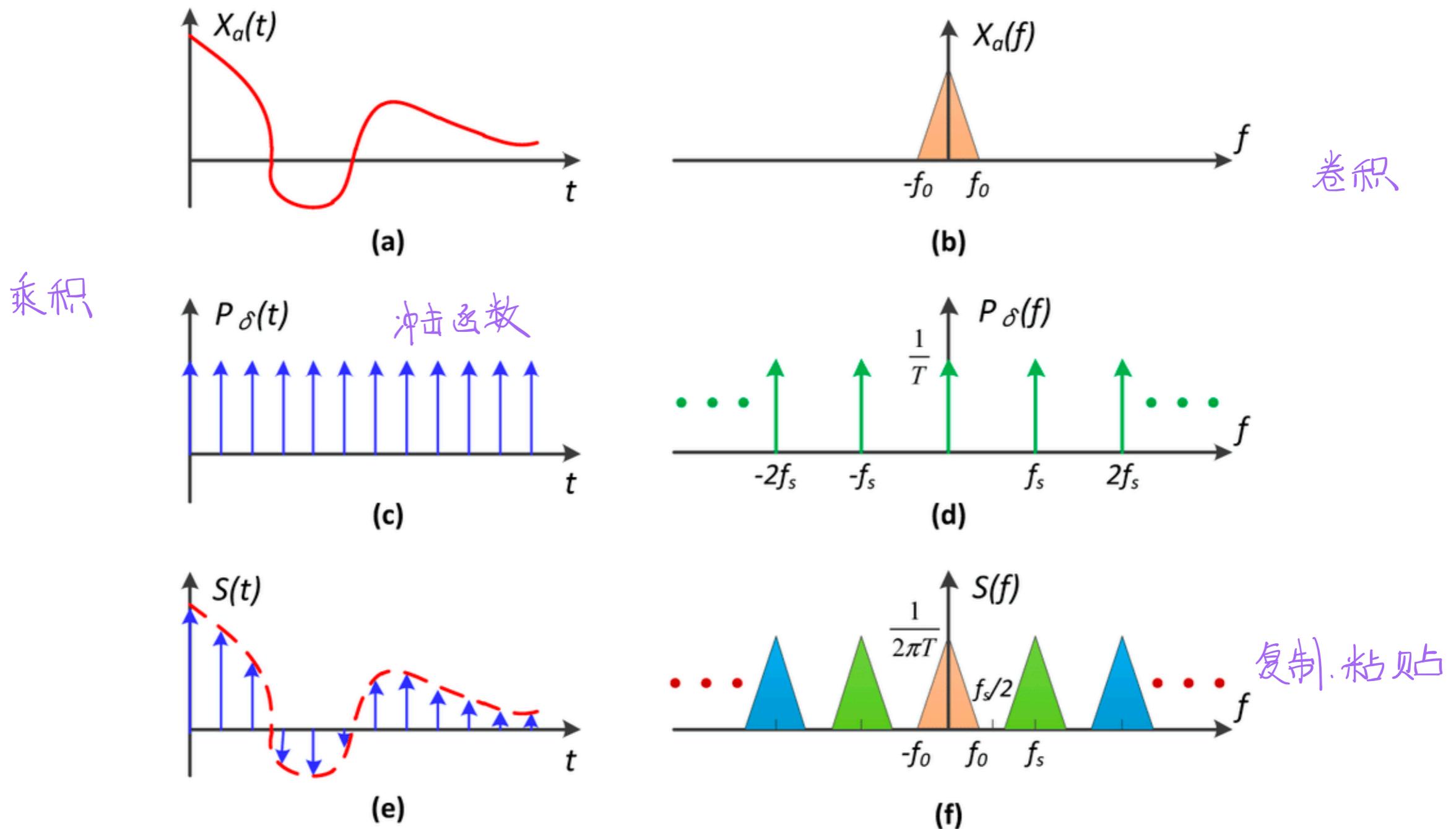


# Wider Filter Kernel = Lower Frequencies



采样 Sampling = Repeating  
重复频域上的内容 Frequency Contents

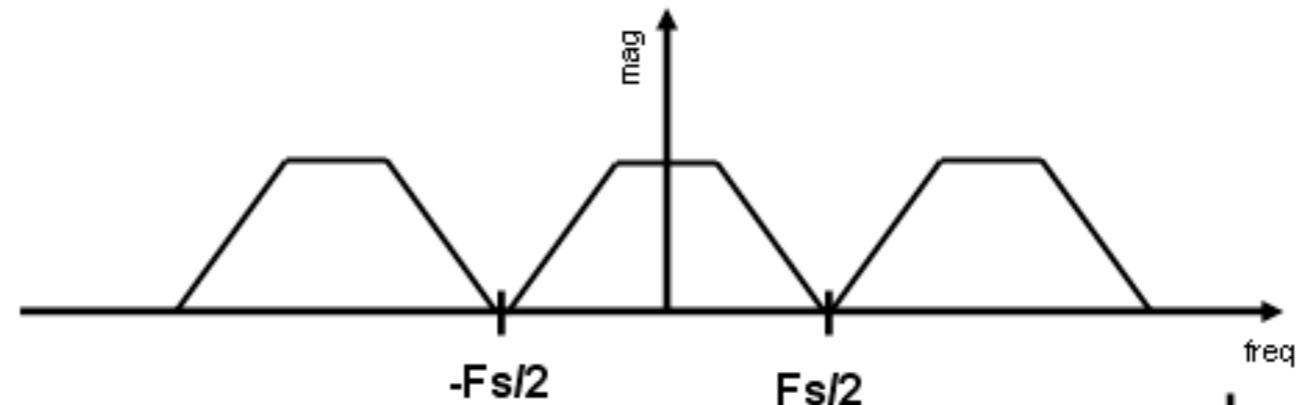
# Sampling = Repeating Frequency Contents



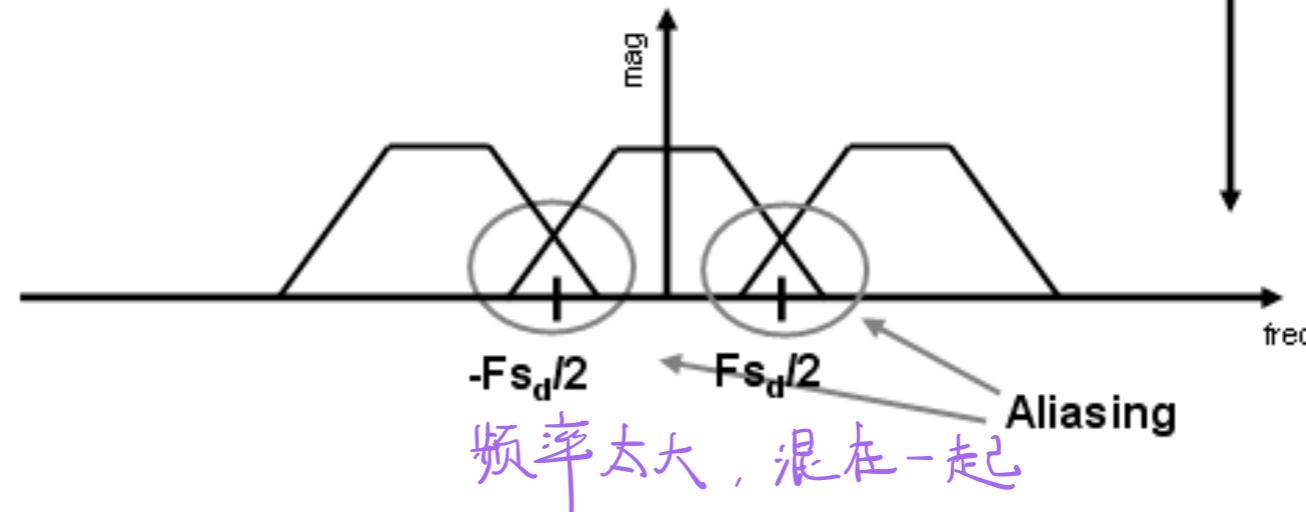
[https://www.researchgate.net/figure/The-evolution-of-sampling-theorem-a-The-time-domain-of-the-band-limited-signal-and-b\\_fig5\\_301556095](https://www.researchgate.net/figure/The-evolution-of-sampling-theorem-a-The-time-domain-of-the-band-limited-signal-and-b_fig5_301556095)

# Aliasing = Mixed Frequency Contents

密集采样  
**Dense sampling:**



**Sparse sampling:**



# Antialiasing

# How Can We Reduce Aliasing Error?

Option 1: Increase sampling rate 提高采样率 (增加分辨率)

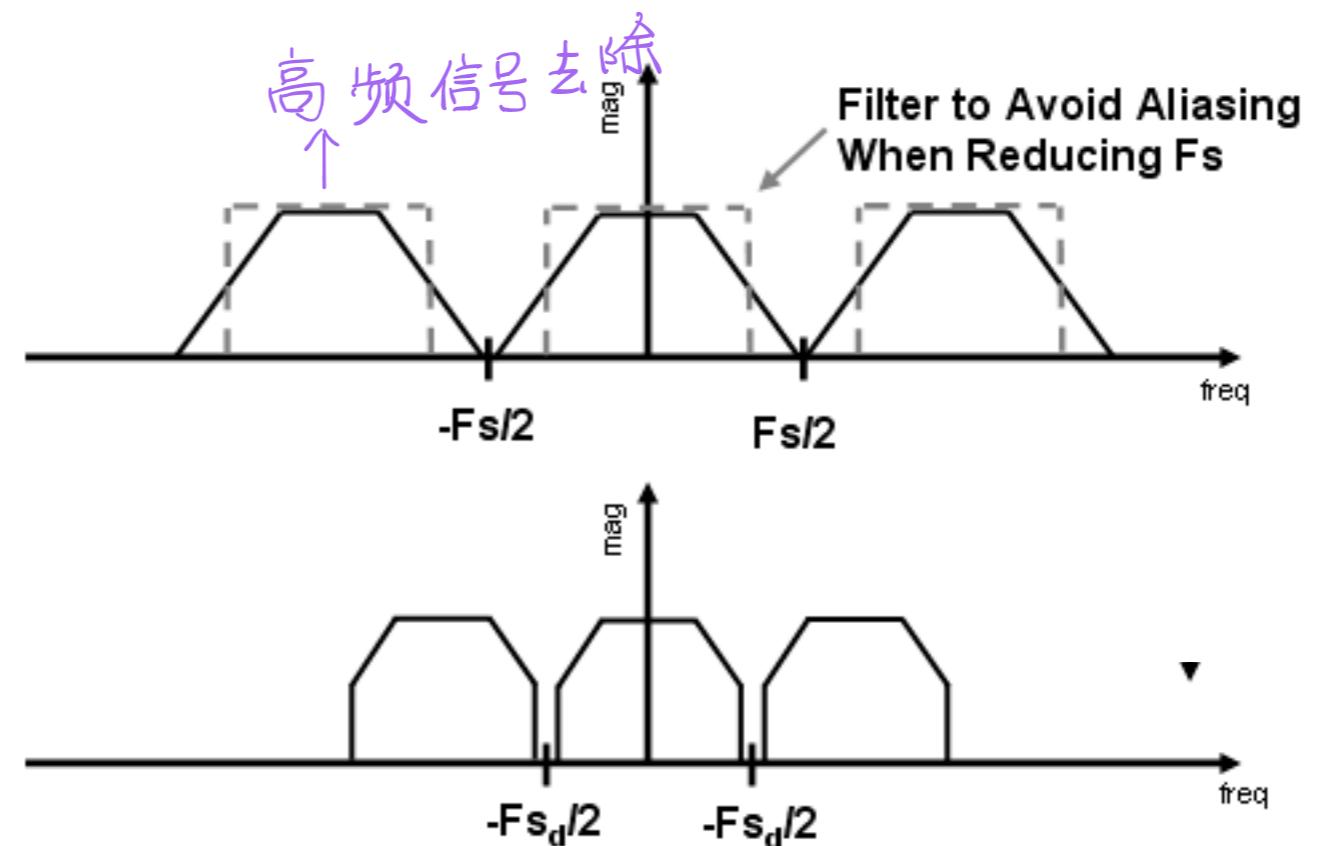
- Essentially increasing the distance between replicas in the Fourier domain
- Higher resolution displays, sensors, framebuffers...
- But: costly & may need very high resolution

Option 2: Antialiasing 反走样

- Making Fourier contents “narrower” before repeating
- i.e. **Filtering out high frequencies before sampling**  
模糊  $\Leftrightarrow$  低通滤波：消除高频信号，再采样

# Antialiasing = Limiting, then repeating

**Filtering**



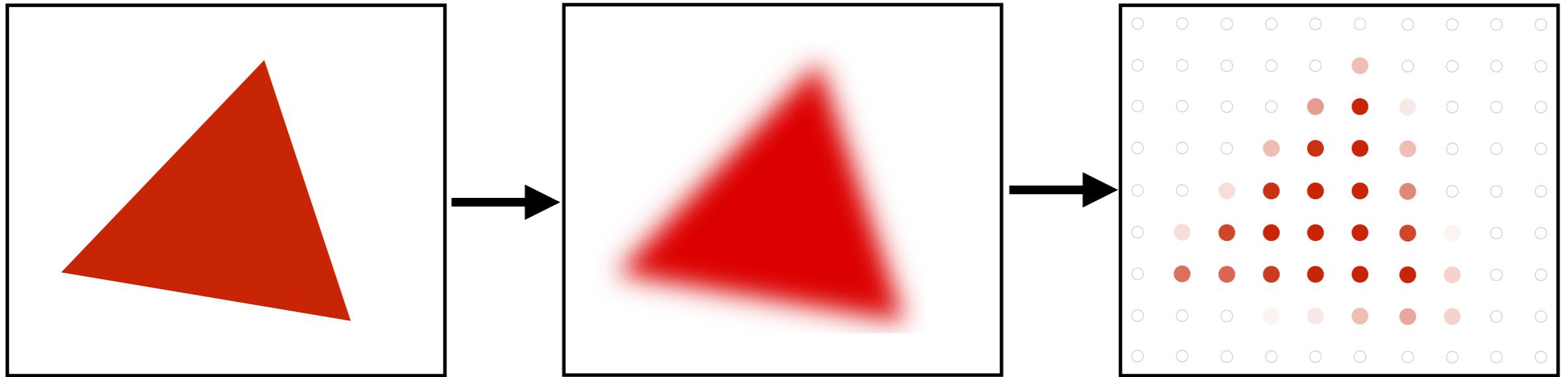
**Then sparse sampling**

# Regular Sampling



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

# Antialiased Sampling



Pre-Filter

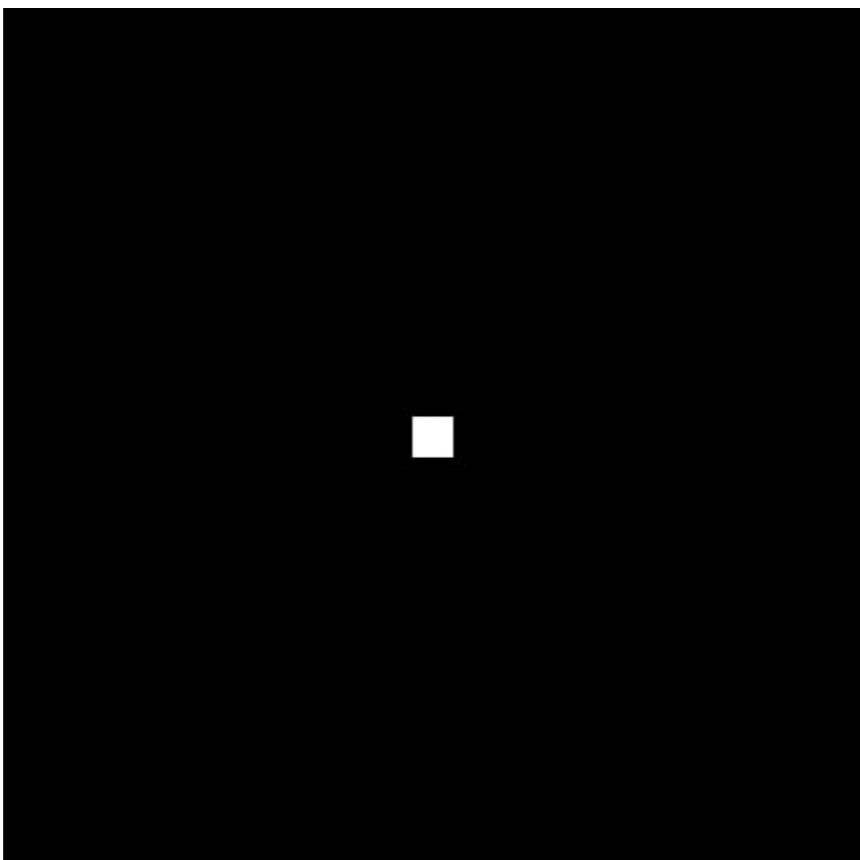
(remove frequencies above Nyquist)

Sample

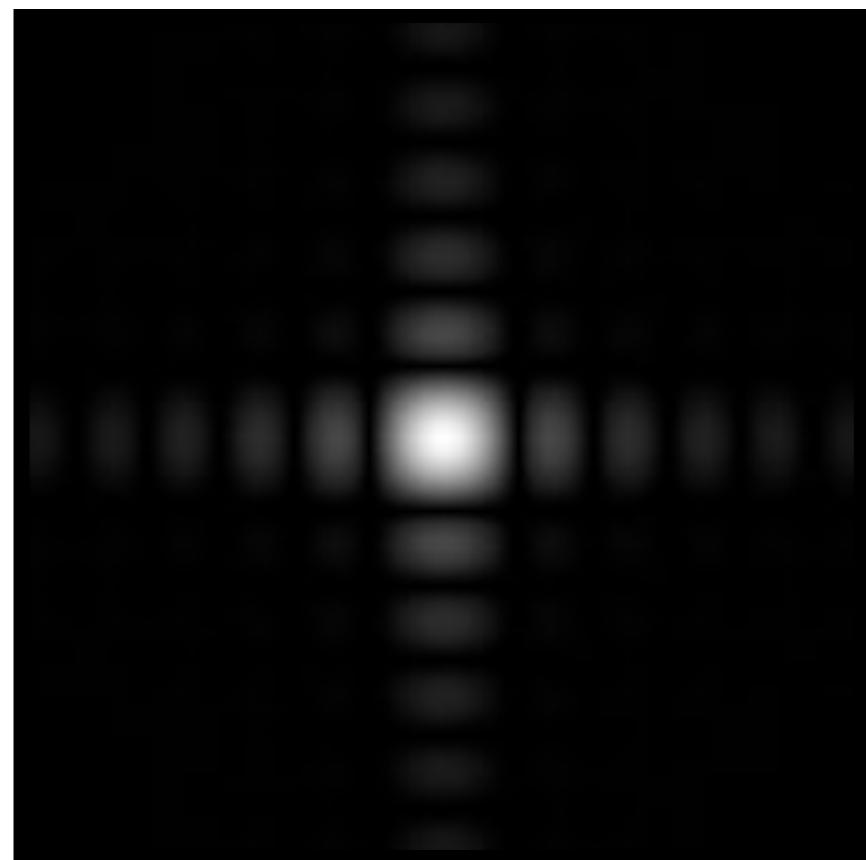
Note antialiased edges in rasterized triangle  
where pixel values take intermediate values

# 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

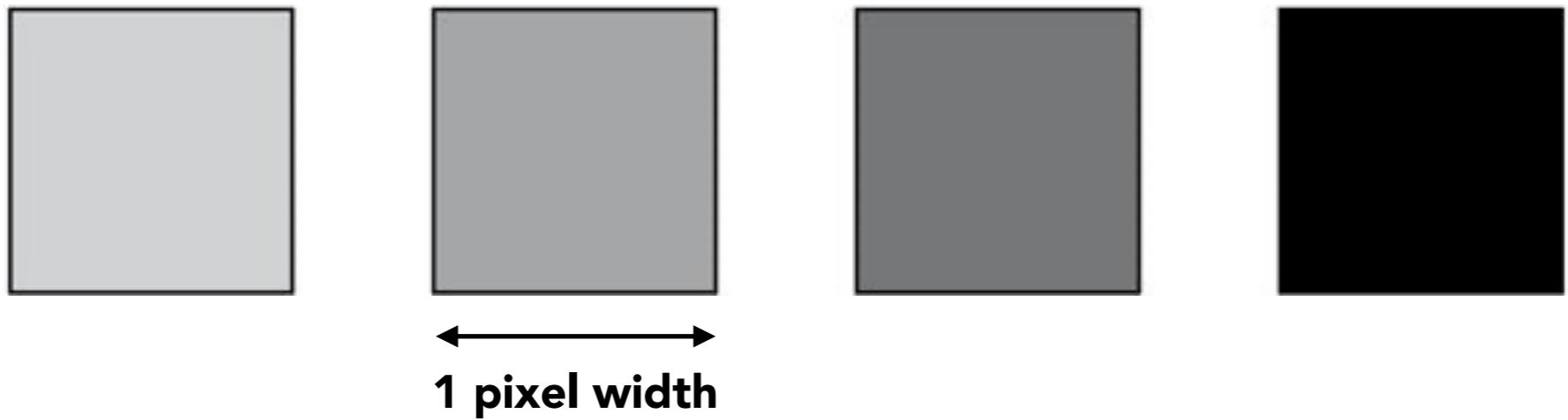
# Antialiasing by Computing Average Pixel Value

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

Original



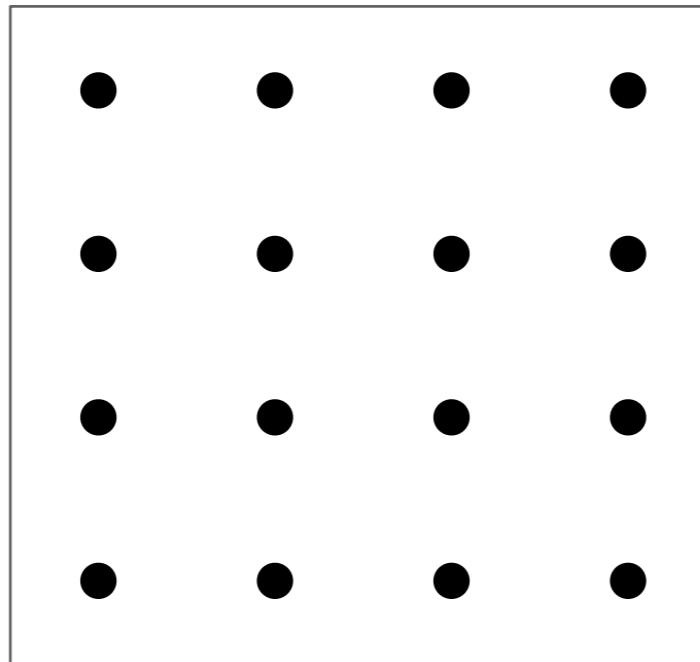
Filtered



# Antialiasing By Supersampling (MSAA)

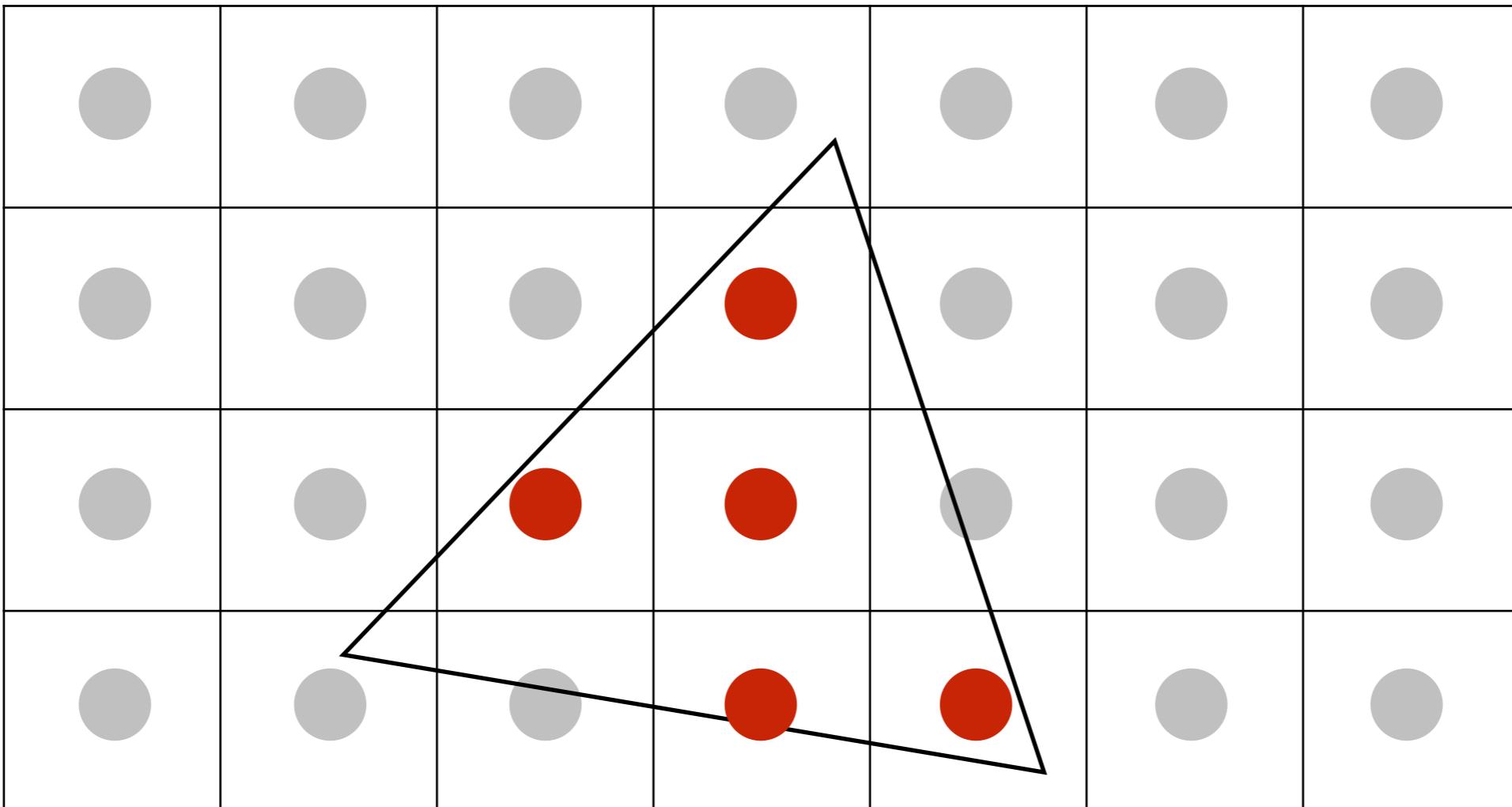
# Supersampling

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



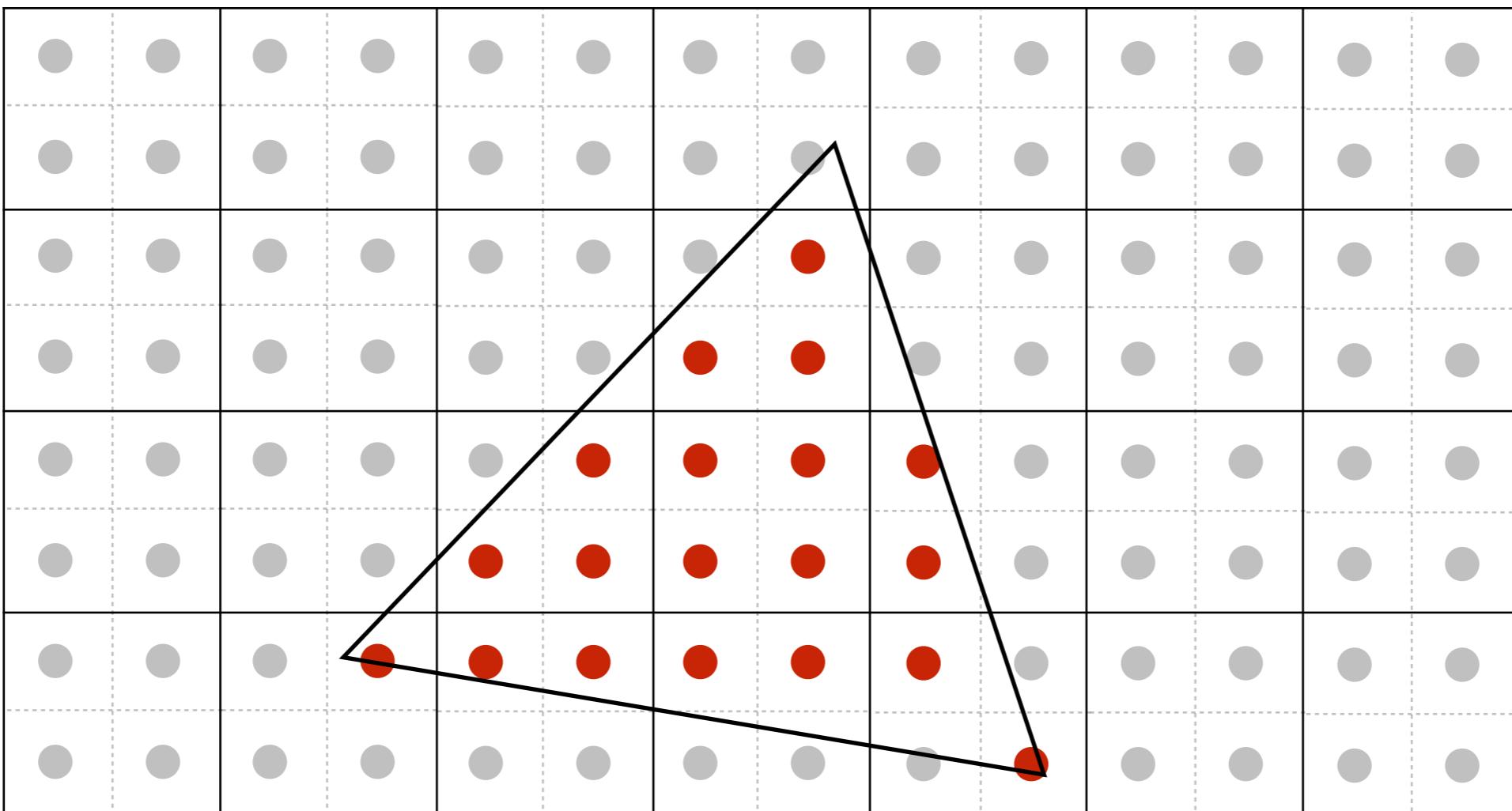
4x4 supersampling

# Point Sampling: One Sample Per Pixel



# Supersampling: Step 1

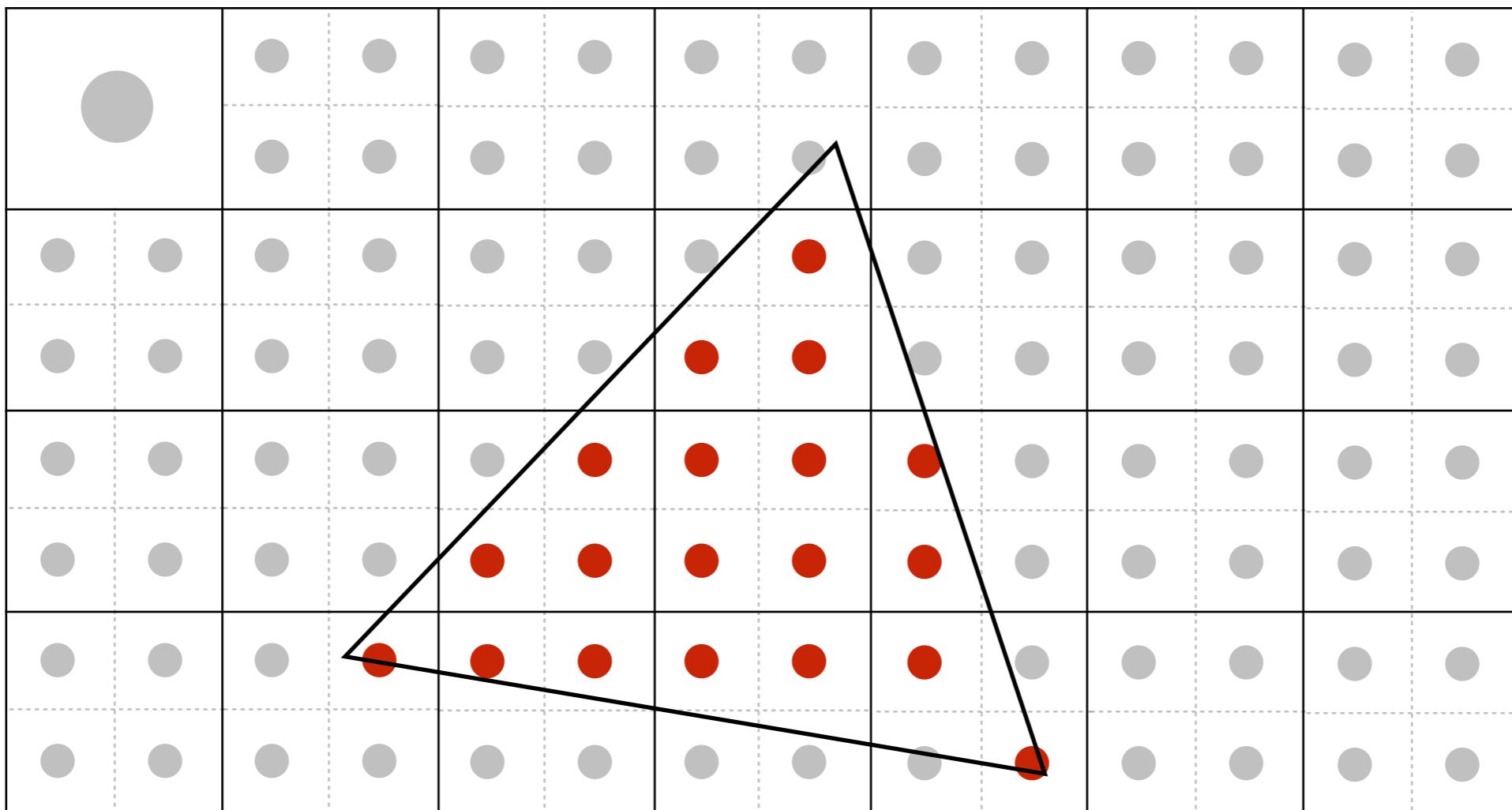
Take NxN samples in each pixel.



2x2 supersampling

# Supersampling: Step 2

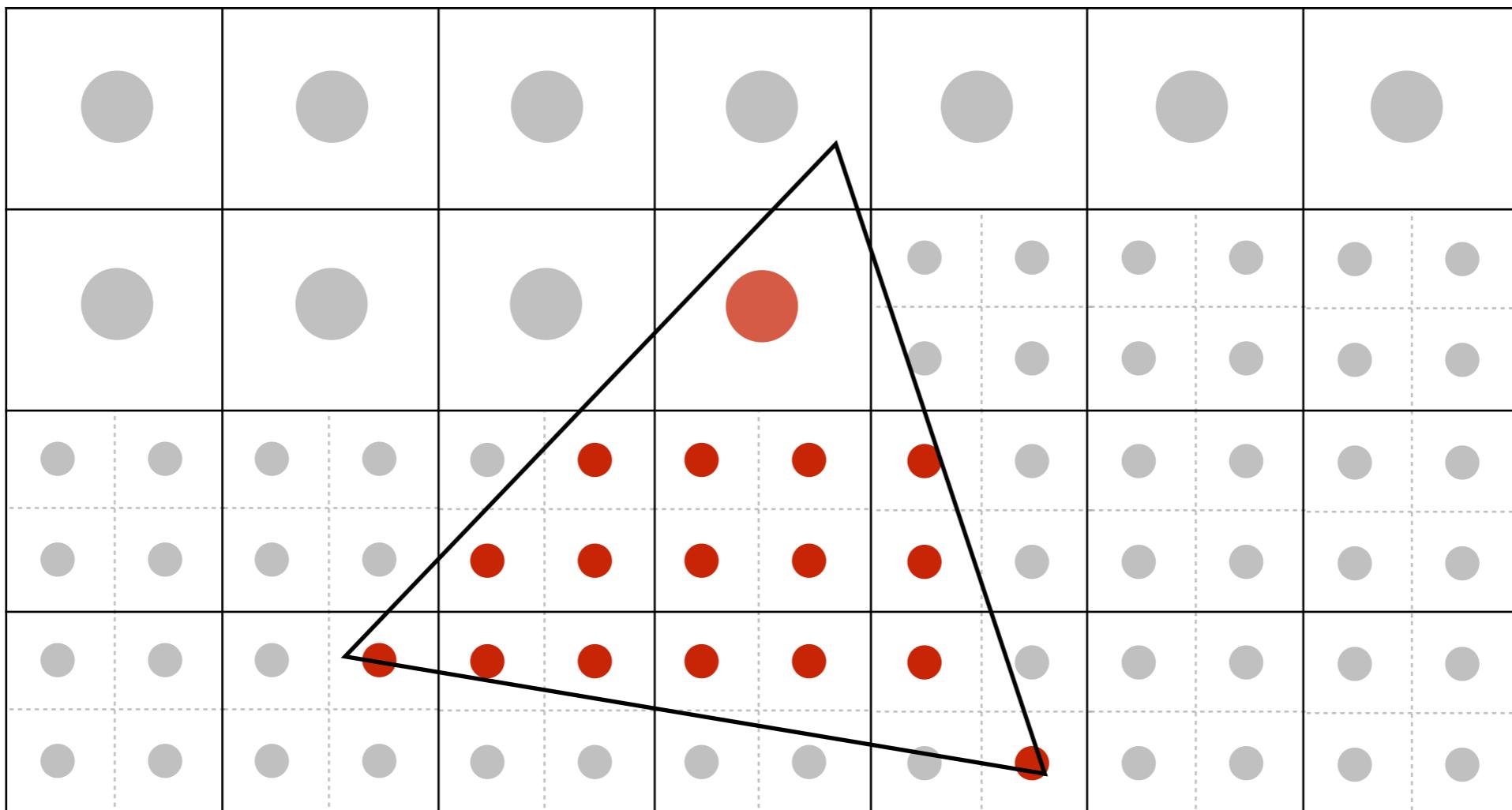
Average the NxN samples “inside” each pixel.



Averaging down

# Supersampling: Step 2

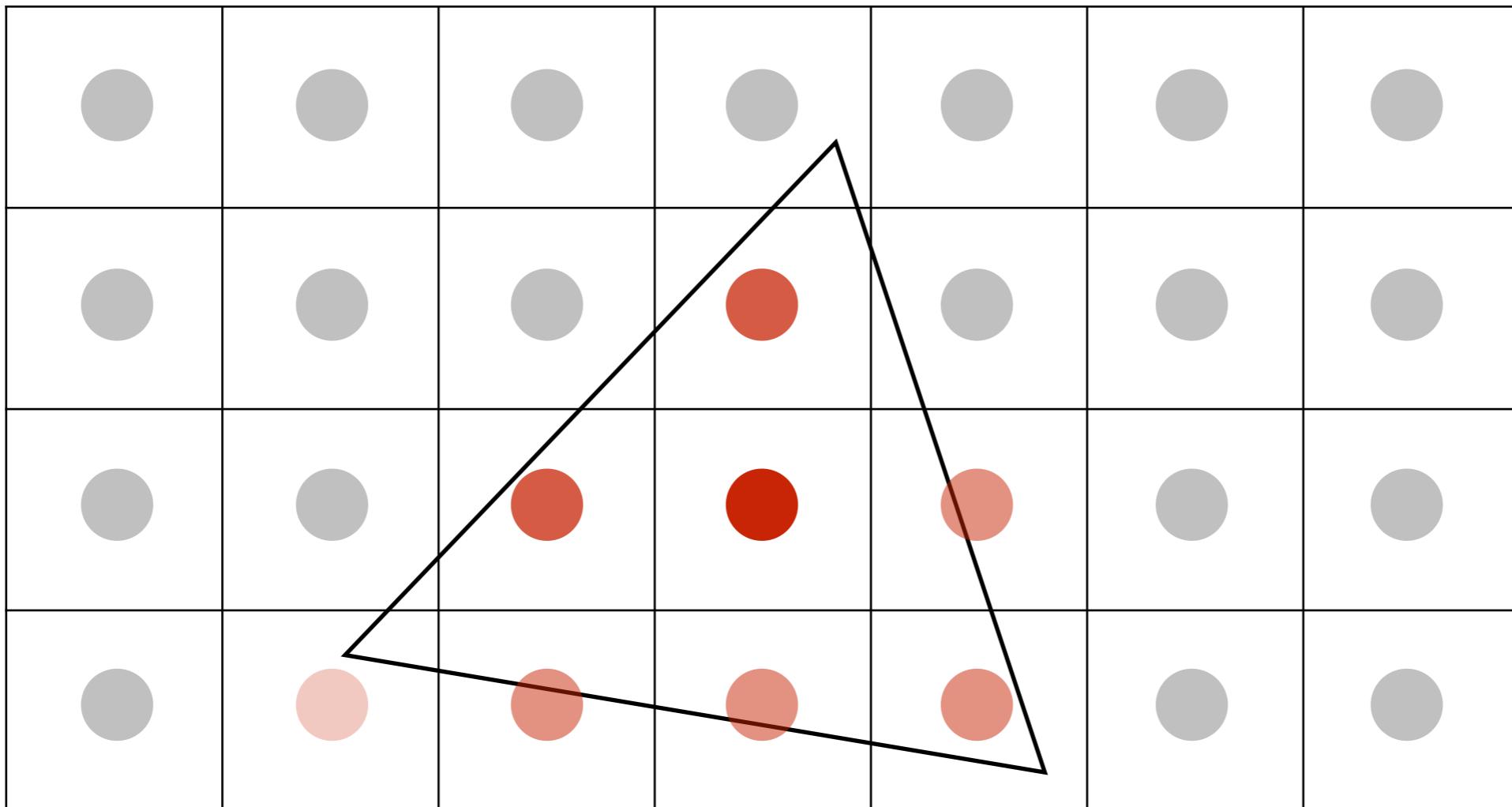
Average the NxN samples “inside” each pixel.



Averaging down

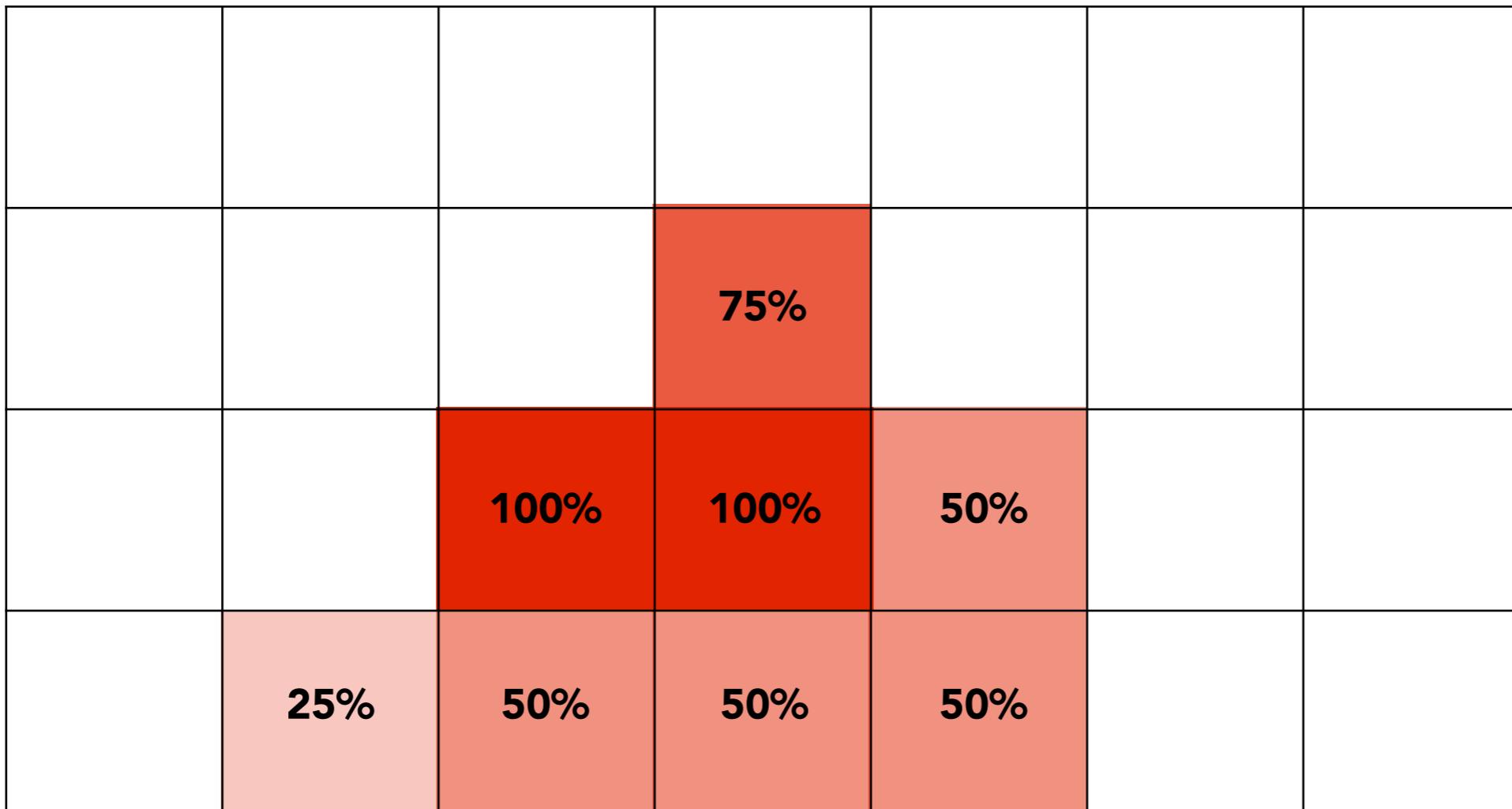
# Supersampling: Step 2

Average the NxN samples “inside” each pixel.

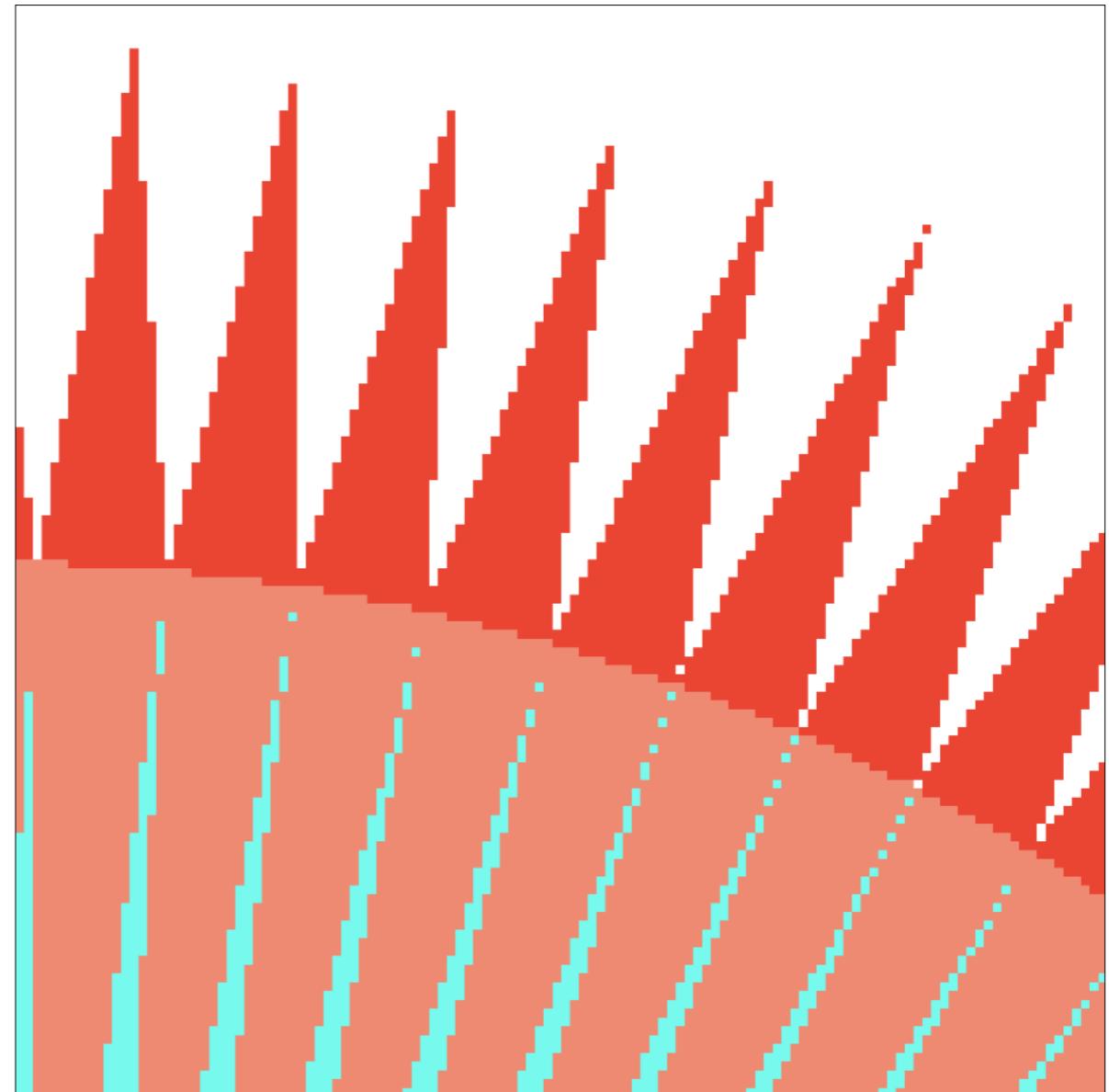
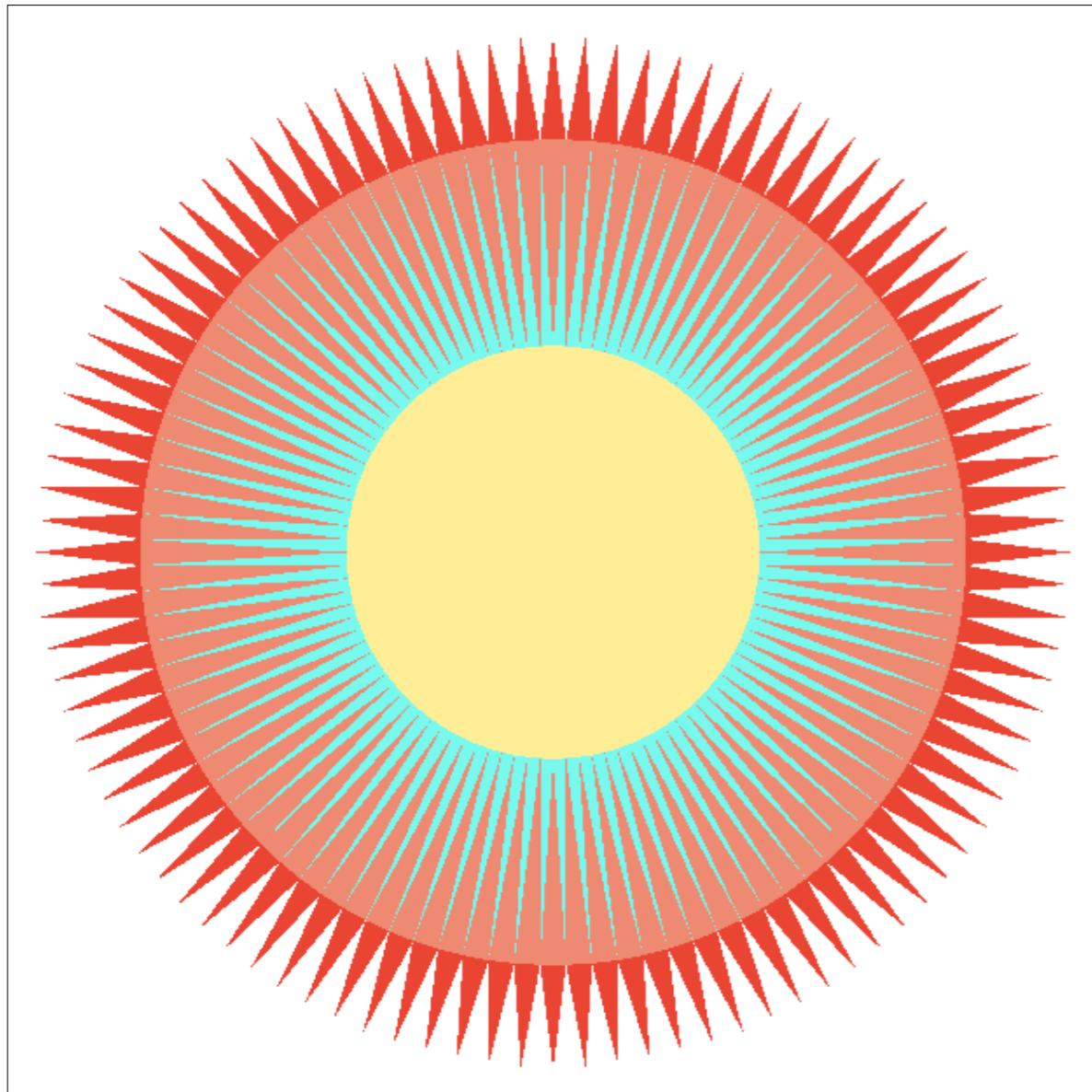


# Supersampling: Result

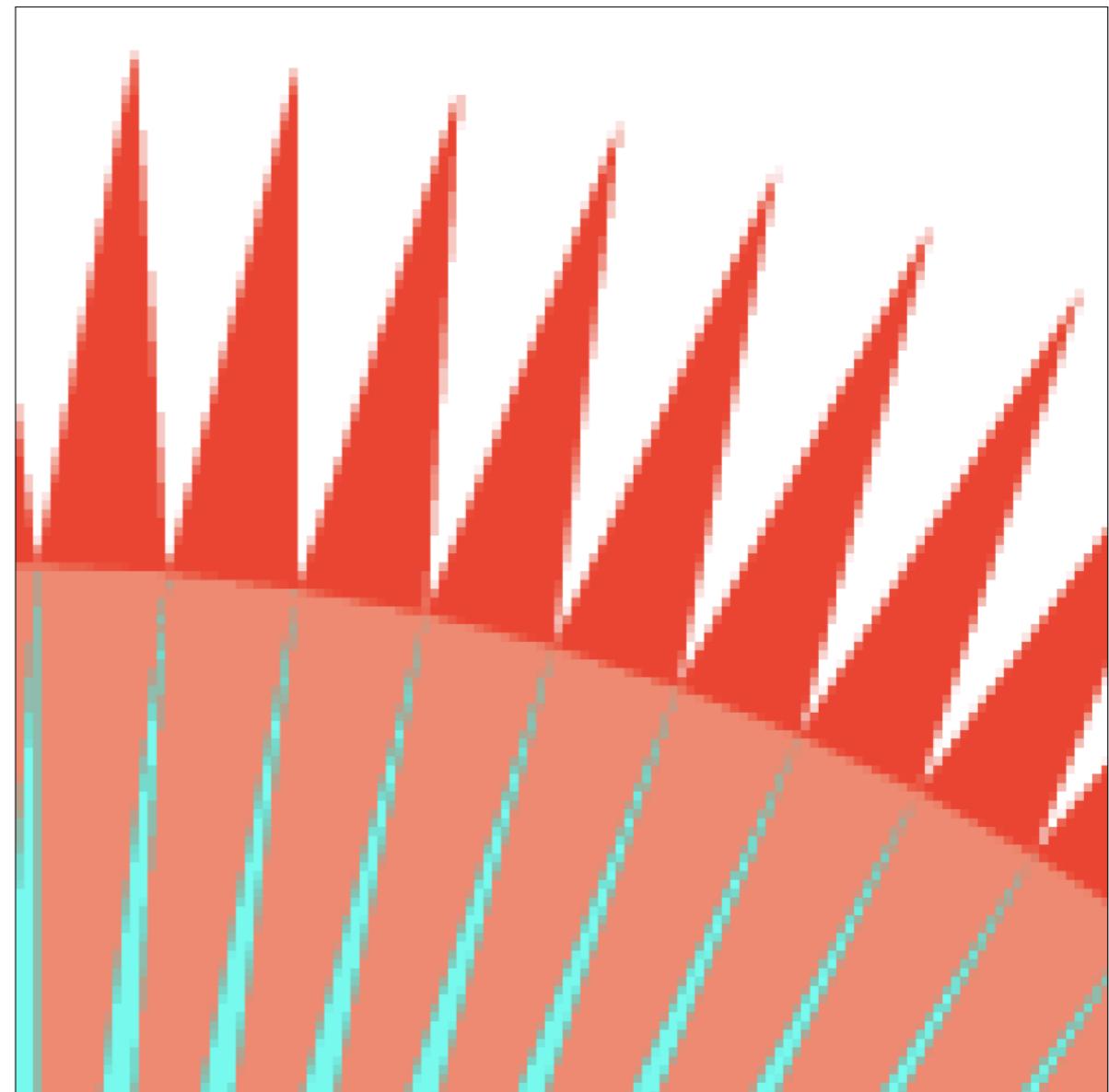
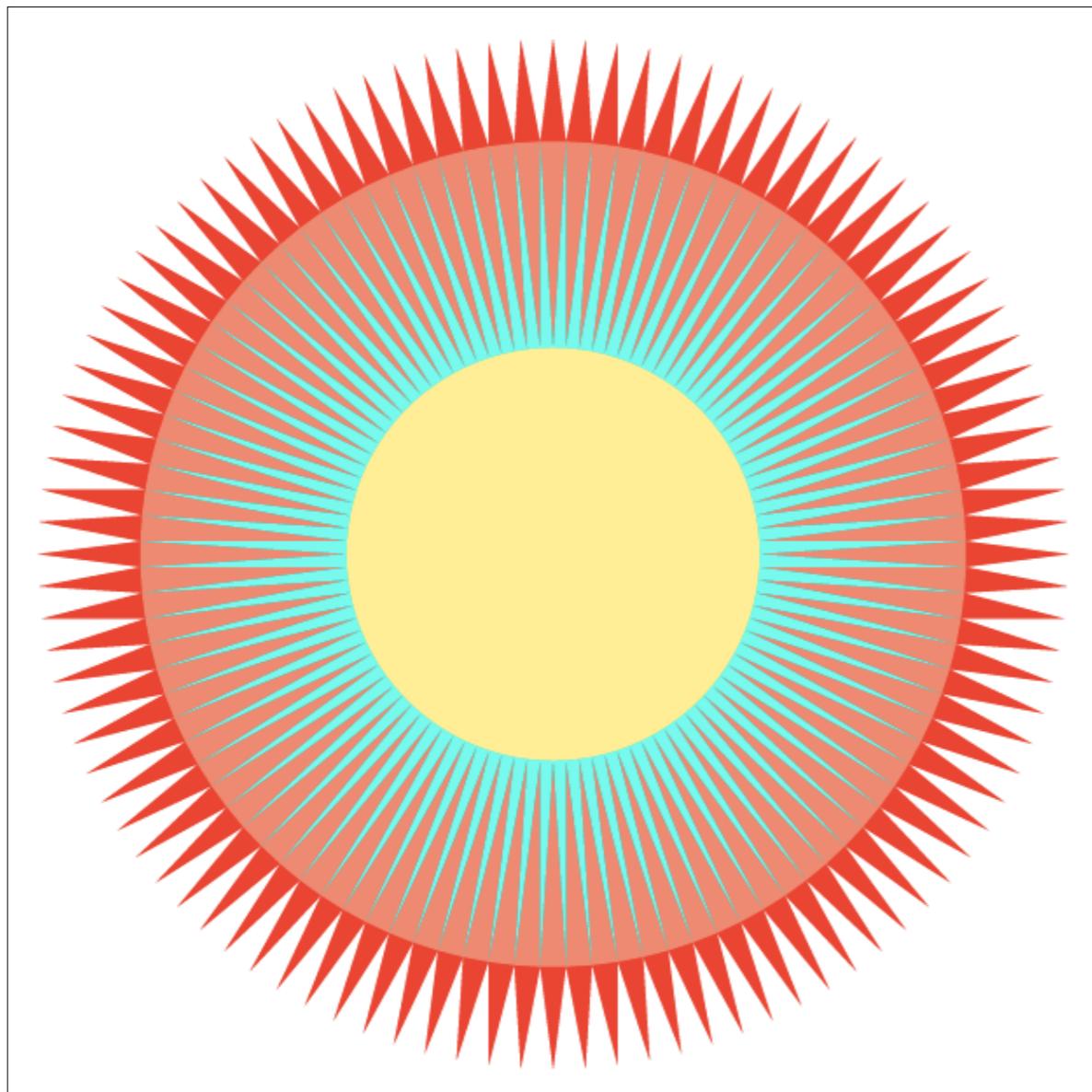
This is the corresponding signal emitted by the display



# Point Sampling



# 4x4 Supersampling



# Antialiasing Today

No free lunch!

- What's the cost of MSAA? 增大计算量

Milestones (personal idea)

- FXAA (Fast Approximate AA) 找到锯齿，更换掉
- TAA (Temporal AA) 复用上一帧

Super resolution / super sampling 超采样（本质相同）  
超分辨率

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

**Thank you!**