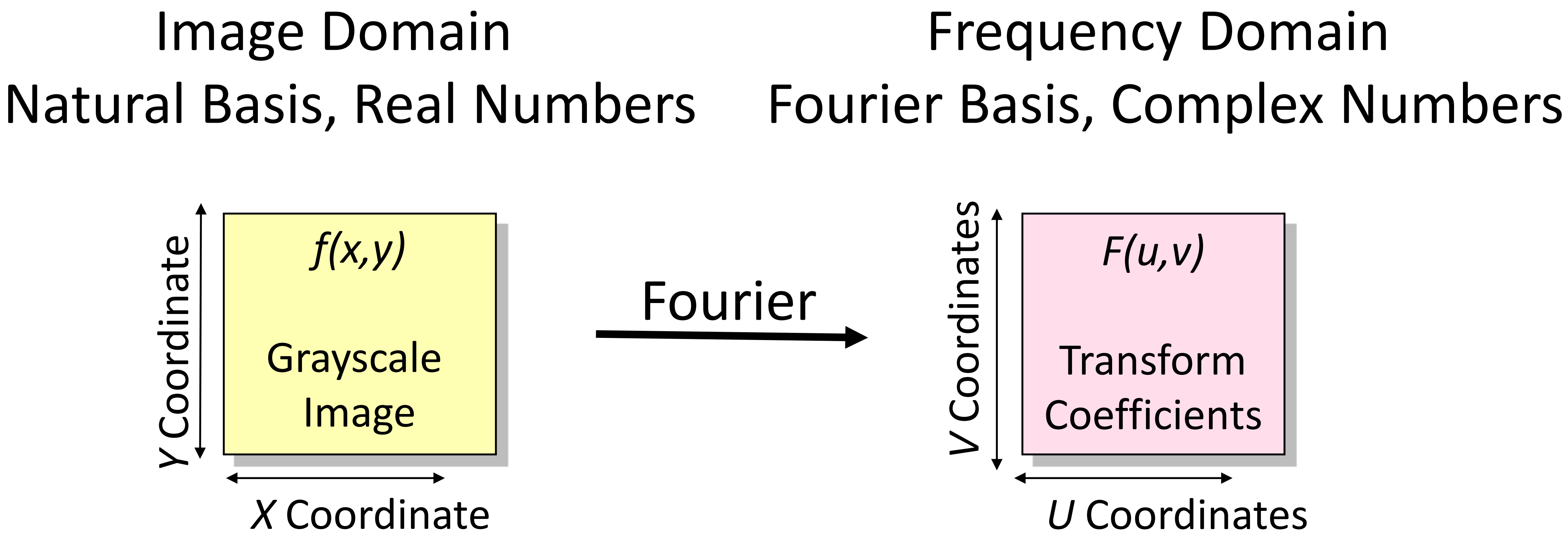
# Fourier Transform of Pictures: Change of Basis in 2D




- The Image & Transform coefficients are arranged in a 2D array.
- Both image $f$ and Fourier transform $F$ are Cyclic / Periodic

# 2D Discrete Fourier

Fourier Transform

$$F(u,v) = \frac{1}{N}\sum_{y=0}^{N-1}\sum_{x=0}^{N-1} f(x,y)\, e^{\frac{-2\pi i(ux+vy)}{N}} \qquad -\infty < \forall\, u,\, v \;> \infty$$

Inverse Fourier Transform

$$f(u,v) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)\, e^{\frac{2\pi i(ux+vy)}{N}} \qquad -\infty < \forall\, x,\, y \;> \infty$$

$$F(0,0) = \frac{1}{N}\sum_{y=0}^{N-1}\sum_{x=0}^{N-1} f(x,y)\, e^{\frac{-2\pi i(0x+0y)}{N}} = \frac{1}{N}\sum_{y=0}^{N-1}\sum_{x=0}^{N-1} f(x,y) = N\bar{f}$$

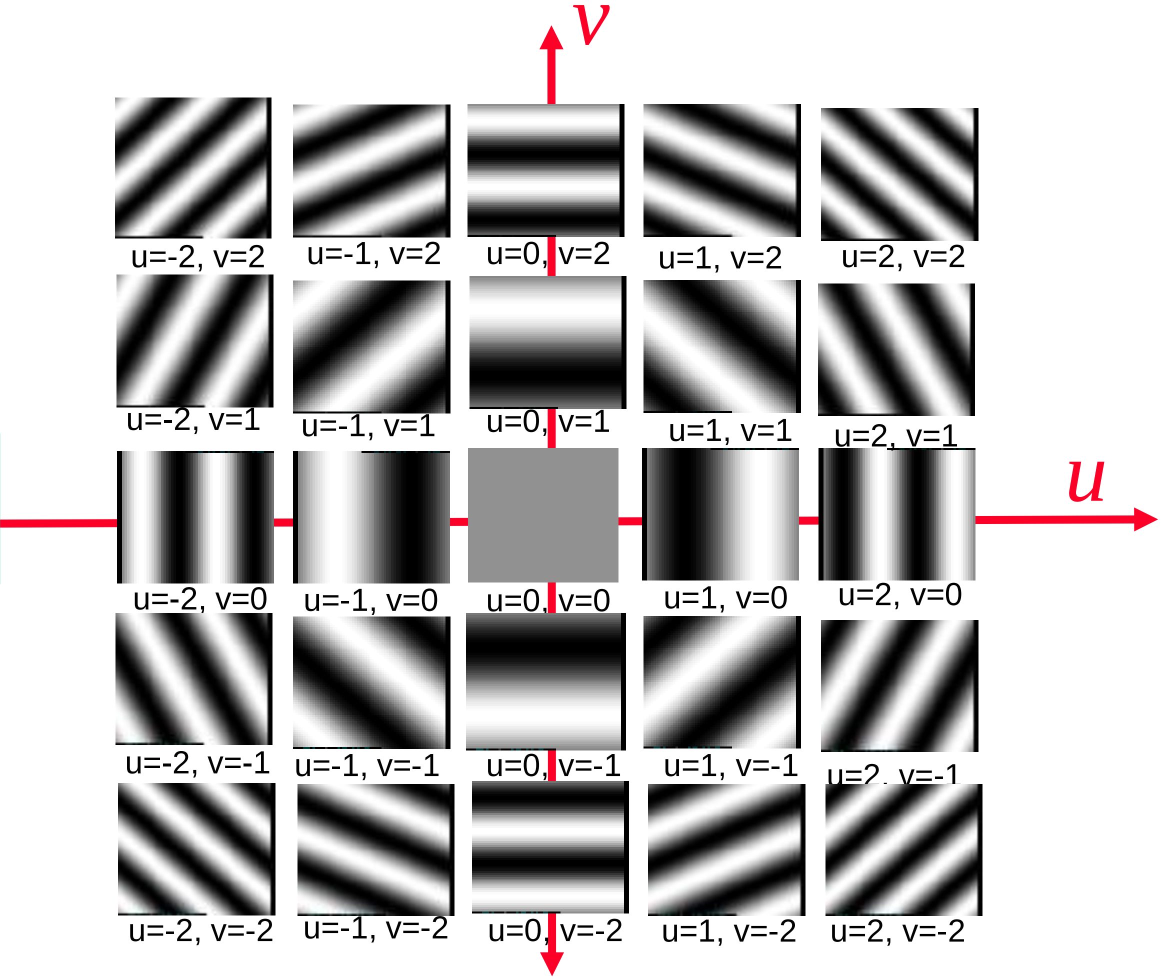
# 2D Fourier Basis Functions

## (For $-2 \leq u, v \leq 2$)

## (Imaginary Part – How can we tell?)

$$e^{\frac{2\pi i(ux+vy)}{N}}$$

$$\sin\left(\frac{2\pi}{N}(ux+vy)\right)$$




The original image is a weighted sum of all basis functions, where the basis function for frequency *(u,v)* is multiplied by the complex number *F(u,v)* specifying:

1. Amplitude
2. Phase (Shift)

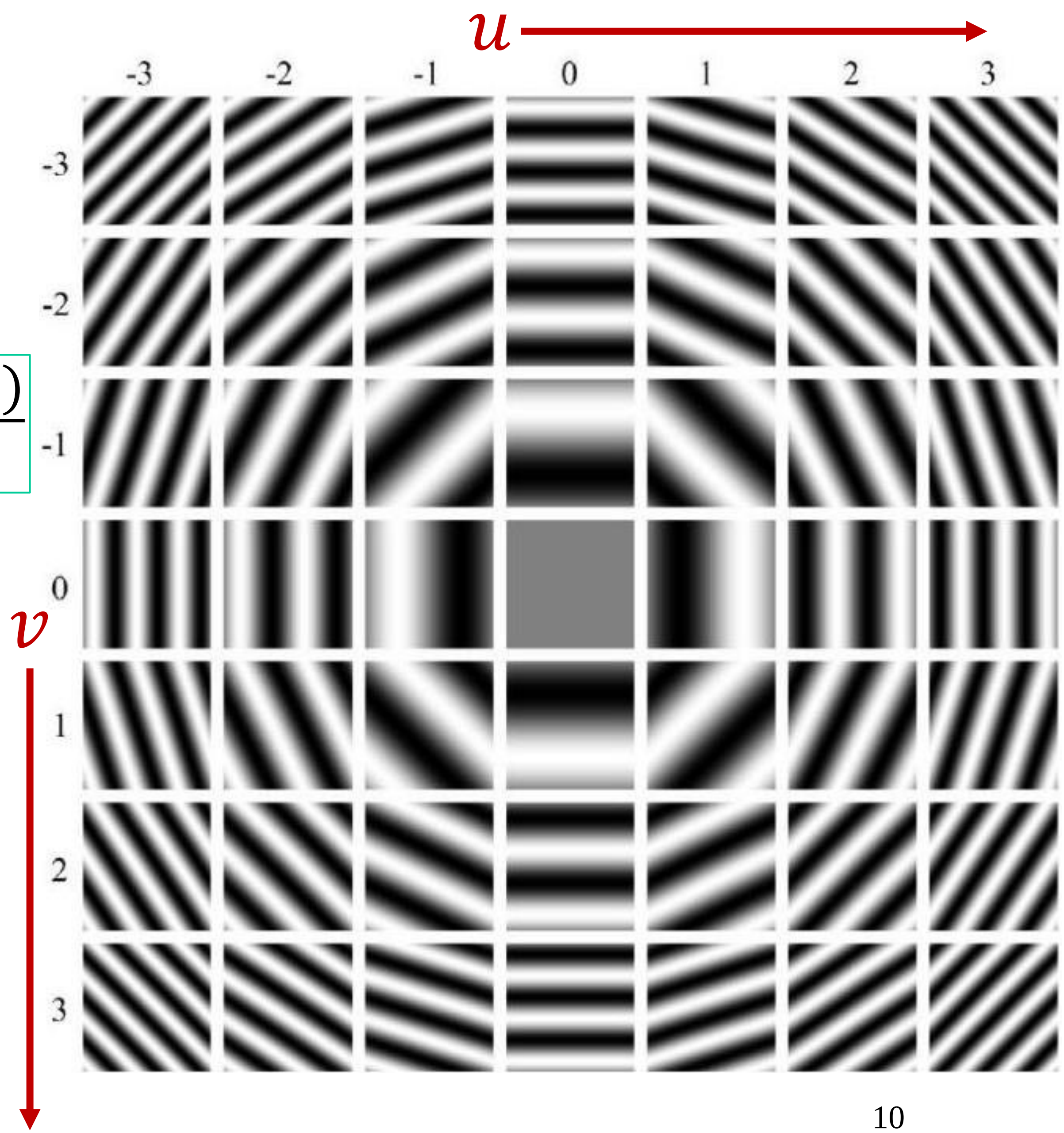
- 25 basis functions for images 5×5.
- Each function is a 5×5 matrix.
- (0,0) at center of each image
- -1 is black, +1 is white, 0 is grey

# 2D Fourier Basis Functions
## (For $-3 \leq u, v \leq 3$)
## (Imaginary Part)

The original image is a weighted sum of all basis functions $\boxed{e^{\frac{2\pi i(ux+vy)}{N}}}$

Showing $\boxed{\sin(\frac{2\pi}{N}(ux + vy))}$

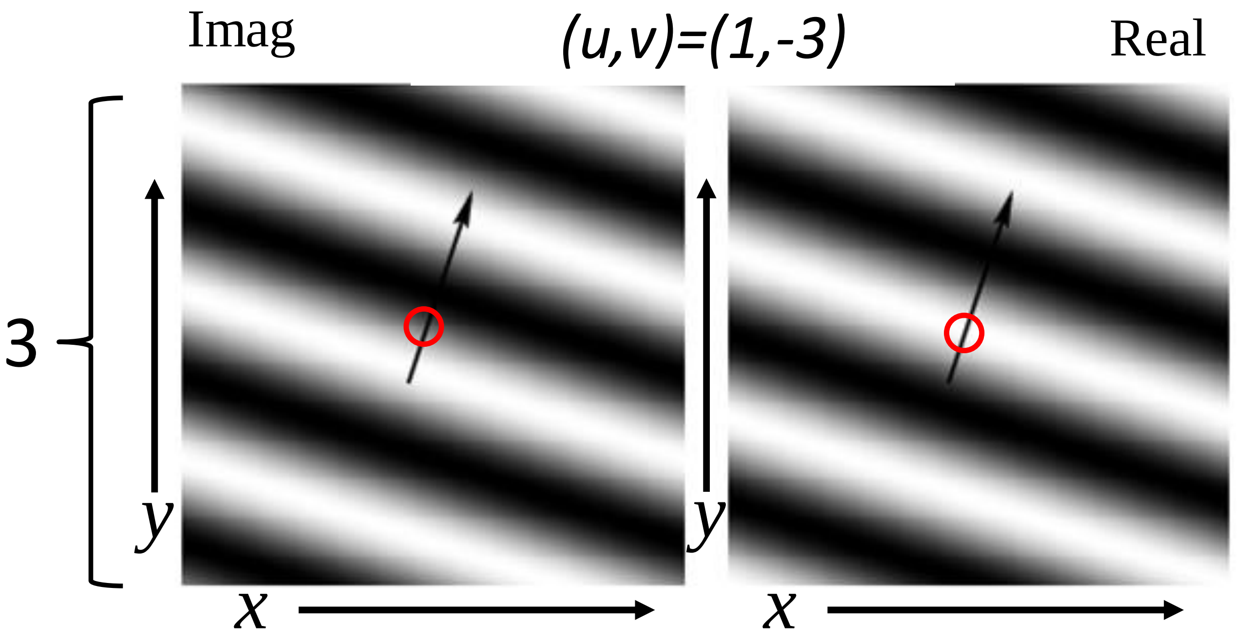
- 49 basis functions (matrices) for all images of size7×7.
-  Each basis function is a 7×7 matrix.
- (0,0) at center of each image
- -1 is black, +1 is white, 0 is grey

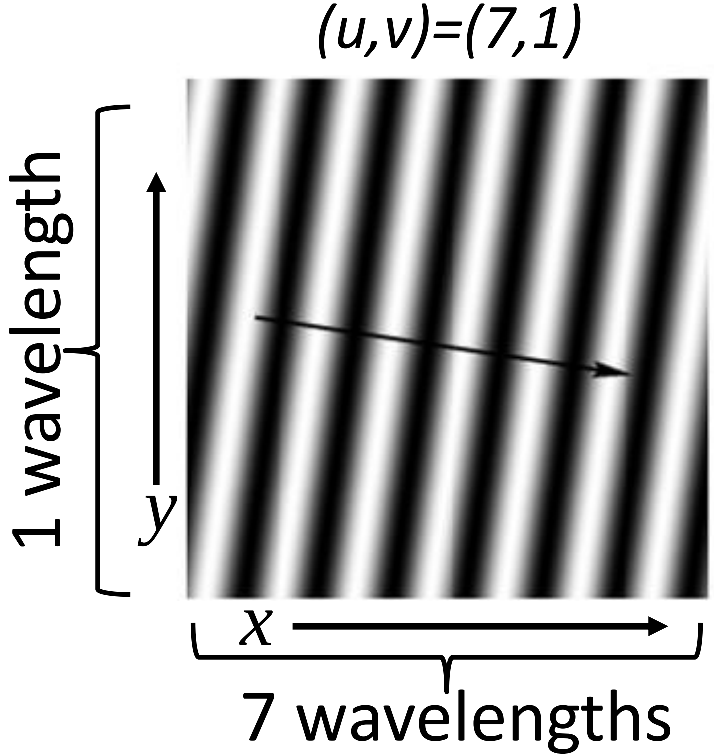


$$e^{\frac{2\pi i(ux+vy)}{N}}$$

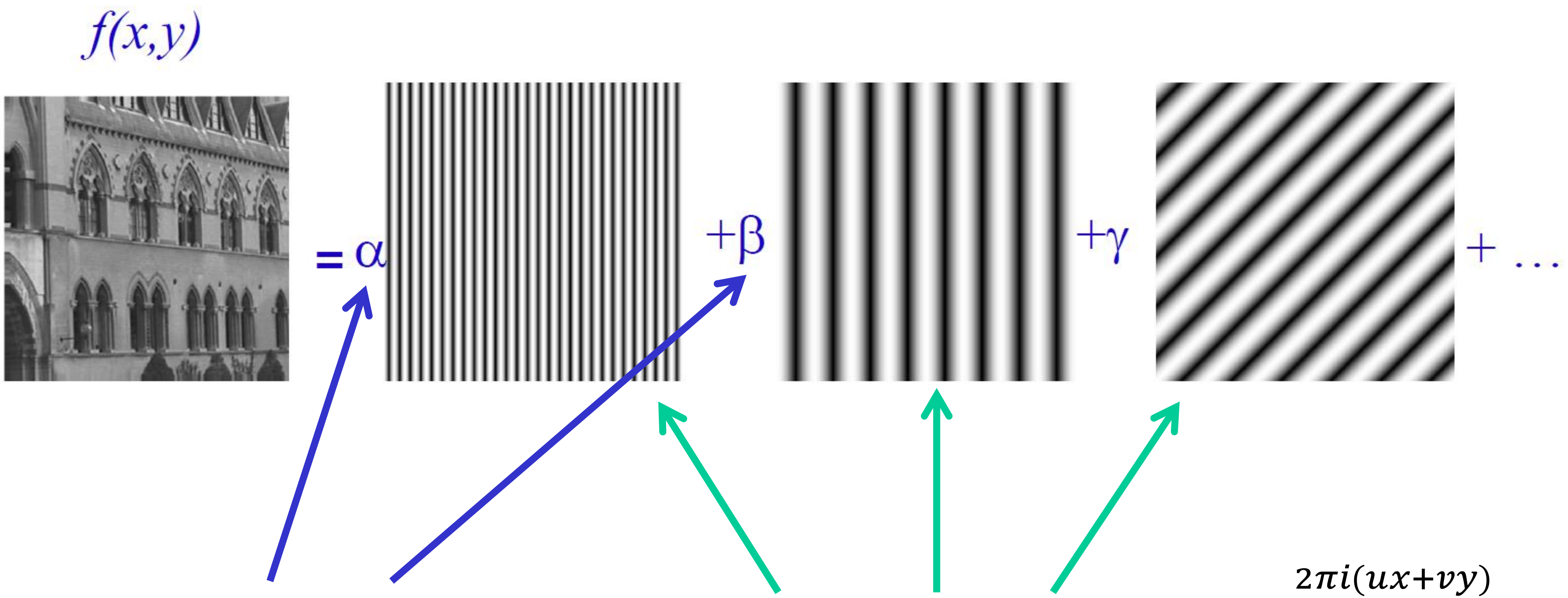
○ Is (0,0)

at center of image



# Summary




Represent complex
Fourier Coefficient $F(u, v)$

Represent complex basis functions $e^{\frac{2\pi i(ux+vy)}{N}}$

# Fourier Spectrum

Fourier Coefficient (complex number): $\boxed{F(u) = R(u) + iI(u)}$

Fourier Spectrum (positive number) $\boxed{|F(u)| = \sqrt{R^2(u) + I^2(u)}}$

Fourier Phase (Angle) $\boxed{\theta(u) = \tan^{-1}(I(u)/R(u))}$

Fourier Coefficient (complex number): $\boxed{F(u) = |F(u)| \ \ e^{i\theta}}$

# Display Fourier Spectrum as Picture

1. Compute $\log(|F(u)| + 1)$

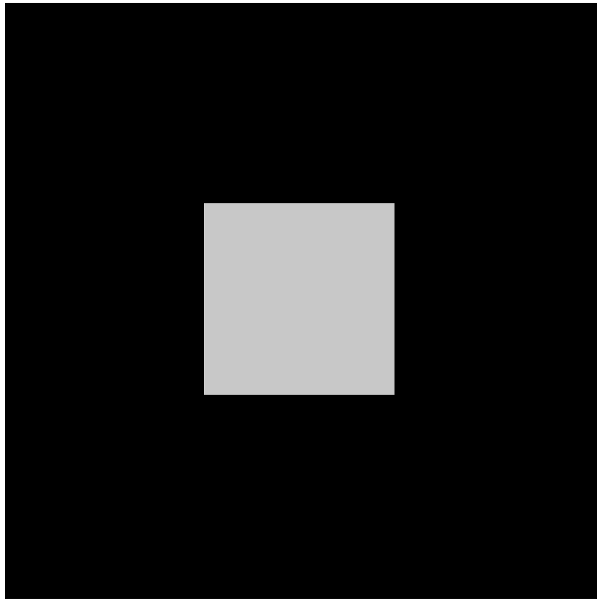
2. Scale to full grey-level range (E.g. 0..255 or 0..1)

3. Move ($u=0,v=0$) to center of image (Shift by $N/2$)

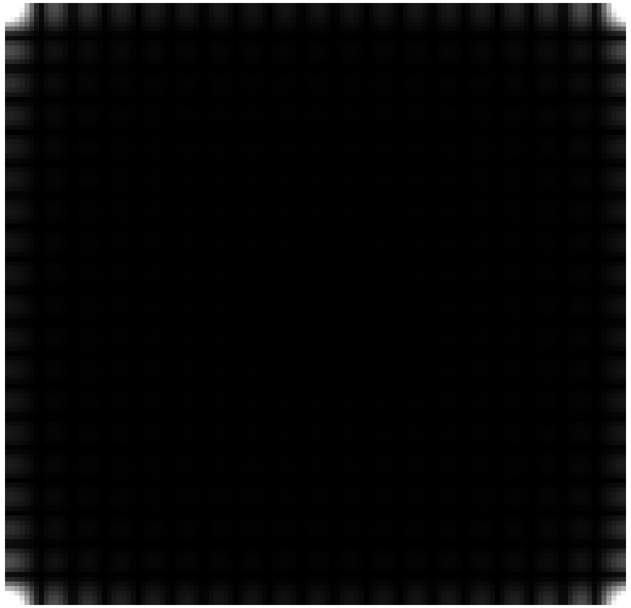
## Log Scale of [0..100]

| | | | | | |
|---|---|---|---|---|---|
| Original *F* | 100 | 4 | 2 | 1 | 0 |
| Log (*1+ F* ) | 4.62 | 1.61 | 1.01 | 0.69 | 0 |
| Scaled to 100 | 100 | 35 | 22 | 15 | 0 |

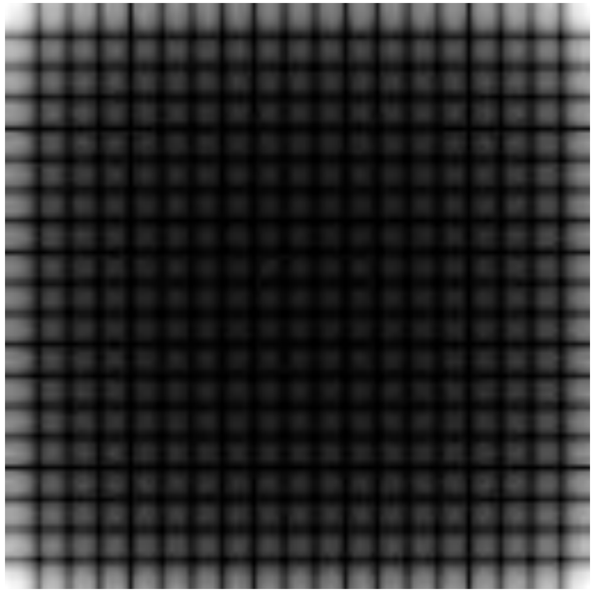
# Display Fourier Spectrum (0 is black)



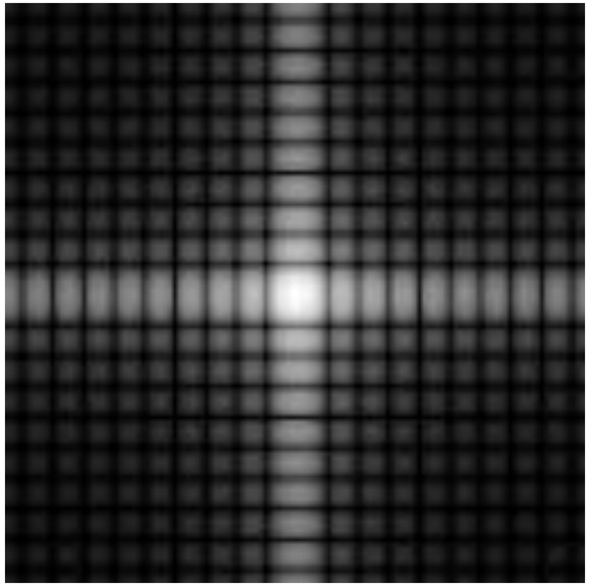
Original Picture



$|F(u,v)|$
(u=0, v=0) in top left



$\log(1 + |F(u,v)|)$
(u=0, v=0) in top left



Shift (u=0, v=0) to center

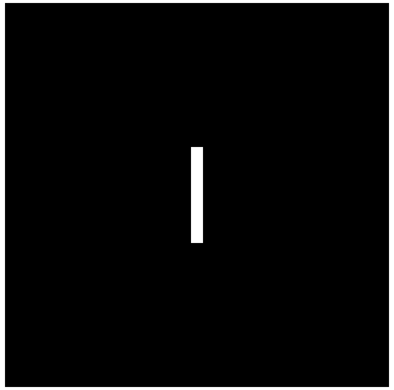
- Question: Why does $|F(0,0)|$ always have the highest value in the Fourier spectrum of an image?
- Hint: The image intensities are always positive

# Translation

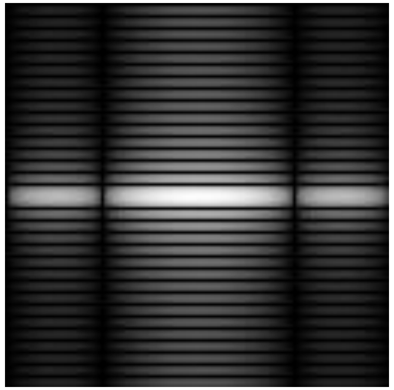
$$f(x-x_0, y-y_0) \Leftrightarrow F(u,v)\, e^{\frac{2\pi i(ux_0+vy_0)}{N}}$$

? $$F(u-u_0, v-v_0) \Leftrightarrow f(x,y)\, e^{\frac{-2\pi i(u_0x+v_0y)}{N}}$$

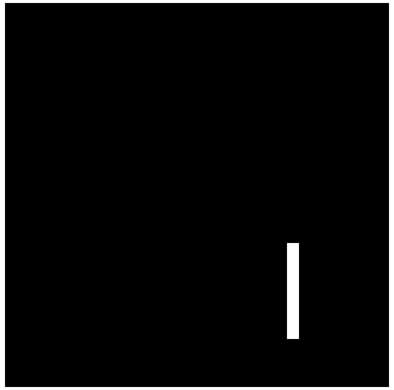
Fourier Spectrum is invariant to image translation



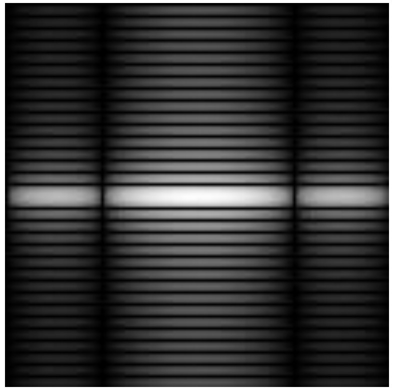
$f$



$|F|$



$f$



$|F|$

# Decomposition Equation

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{\frac{-2\pi i(ux+vy)}{N}}$$

$$e^{\frac{-2\pi i(ux+vy)}{N}} = \left(e^{\frac{-2\pi iux}{N}}\right)\left(e^{\frac{-2\pi ivy}{N}}\right)$$

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} e^{\frac{-2\pi iux}{N}} \sum_{y=0}^{N-1} f(x,y)e^{\frac{-2\pi ivy}{N}}$$

$$F(x,v) = \sum_{y=0}^{N-1} f(x,y)e^{\frac{-2\pi ivy}{N}}$$

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} e^{\frac{-2\pi iux}{N}} F(x,v)$$

– Compute 1-D Fourier on each column

On result:

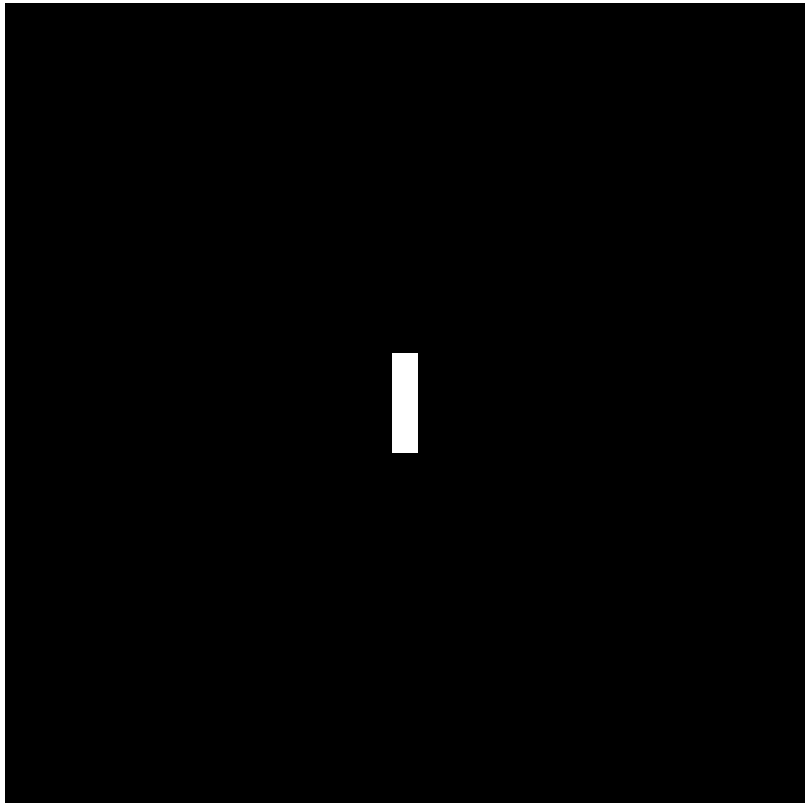
– Compute 1-D Fourier on each row

# Decomposition Conclusions

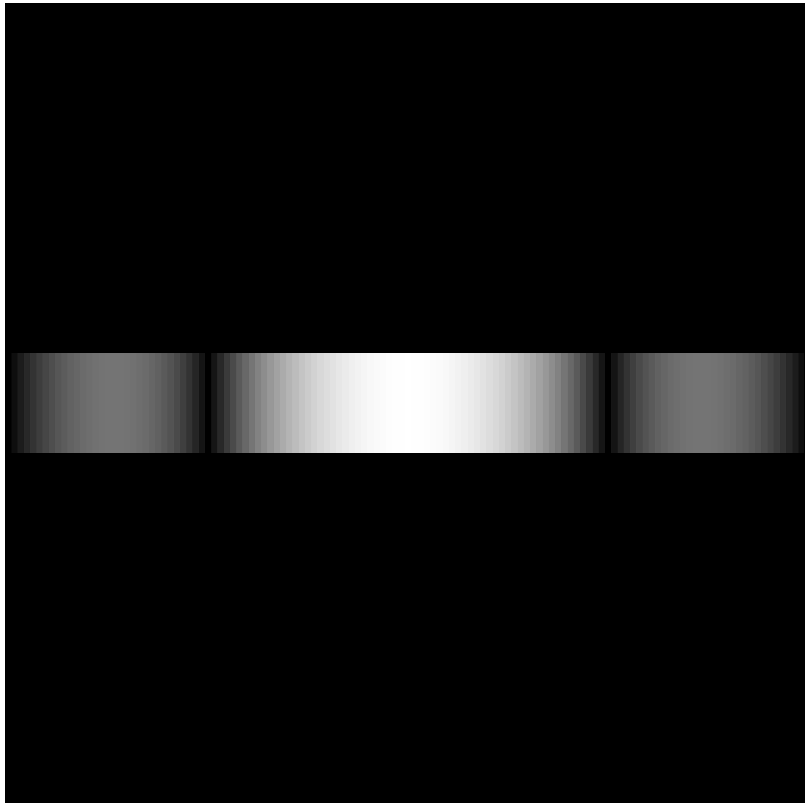
- 2-D Fourier Transform can be computed using 1-D Fourier
  - Compute 1-D Fourier on each column

    On result:
  - Compute 1-D Fourier on each row
  - (Multiply by $N$?)
- 1-D Fourier Transform is enough to compute Fourier of ANY dimension

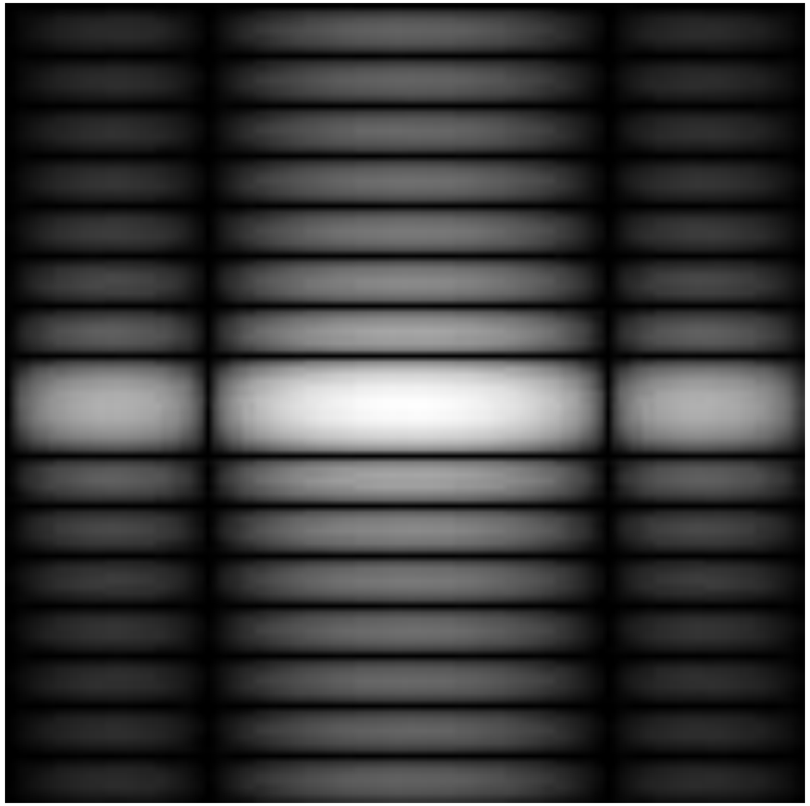
# Decomposition Example



Original
picture



Fourier
in rows



Fourier
in columns

# Periodicity & Symmetry

$$F(u,v) = F(u+N,v) = F(u,v+N) =$$
$$= F(u+N,v+N)$$

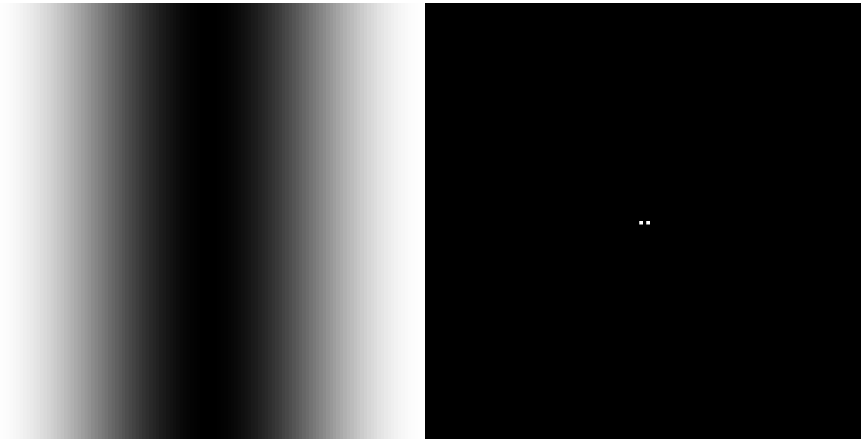
$$F(u,v) = F^*(-u,-v)$$

$$(a+bi)^* = (a-bi)$$

$$|F(u,v)| = |F(-u,-v)|$$

# Fourier Spectrum of Cos($2\pi nx/N$) [$N$ samples]

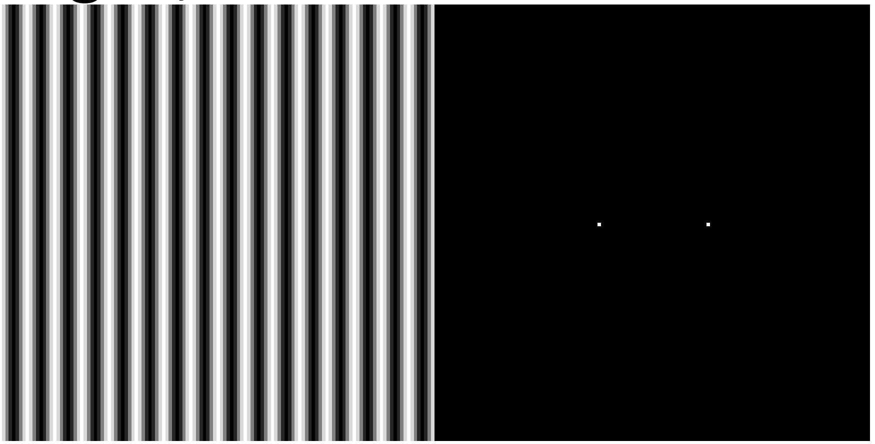
Image1, $n$=1 Fourier1



$|F(-1,0)|=|F(1,0)|=1$

Why 2 points in Fourier Spectrum?

Image2, $n$ =$N/8$ Fourier2



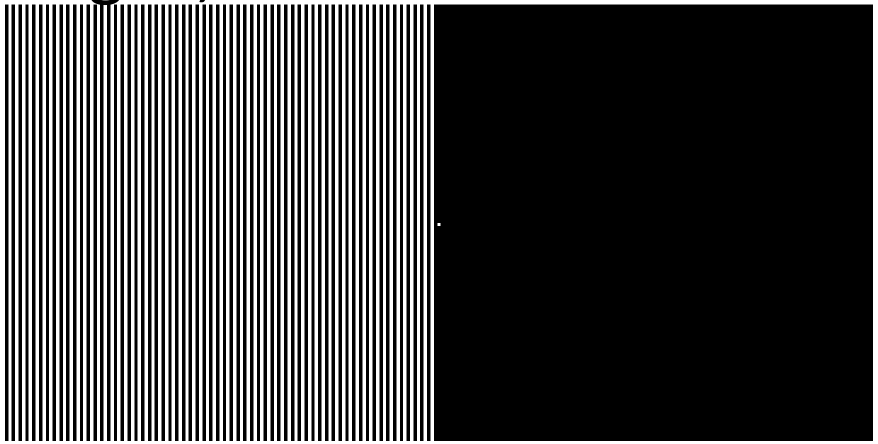
$|F(-8,0)|=|F(8,0)|=1$

Note: Image of Cos has negative values, -1 is black, 1 is white

Cos($\pi x$)

1, -1, 1, -1, …

Image3, $n$=$N/2$ Fourier3



What will be F for 1+Cos($2\pi nx/N$)

$|F(-N/2,0)|=|F(N/2,0)|=1$

# Linearity ($\Phi$ is Transform Fourier)

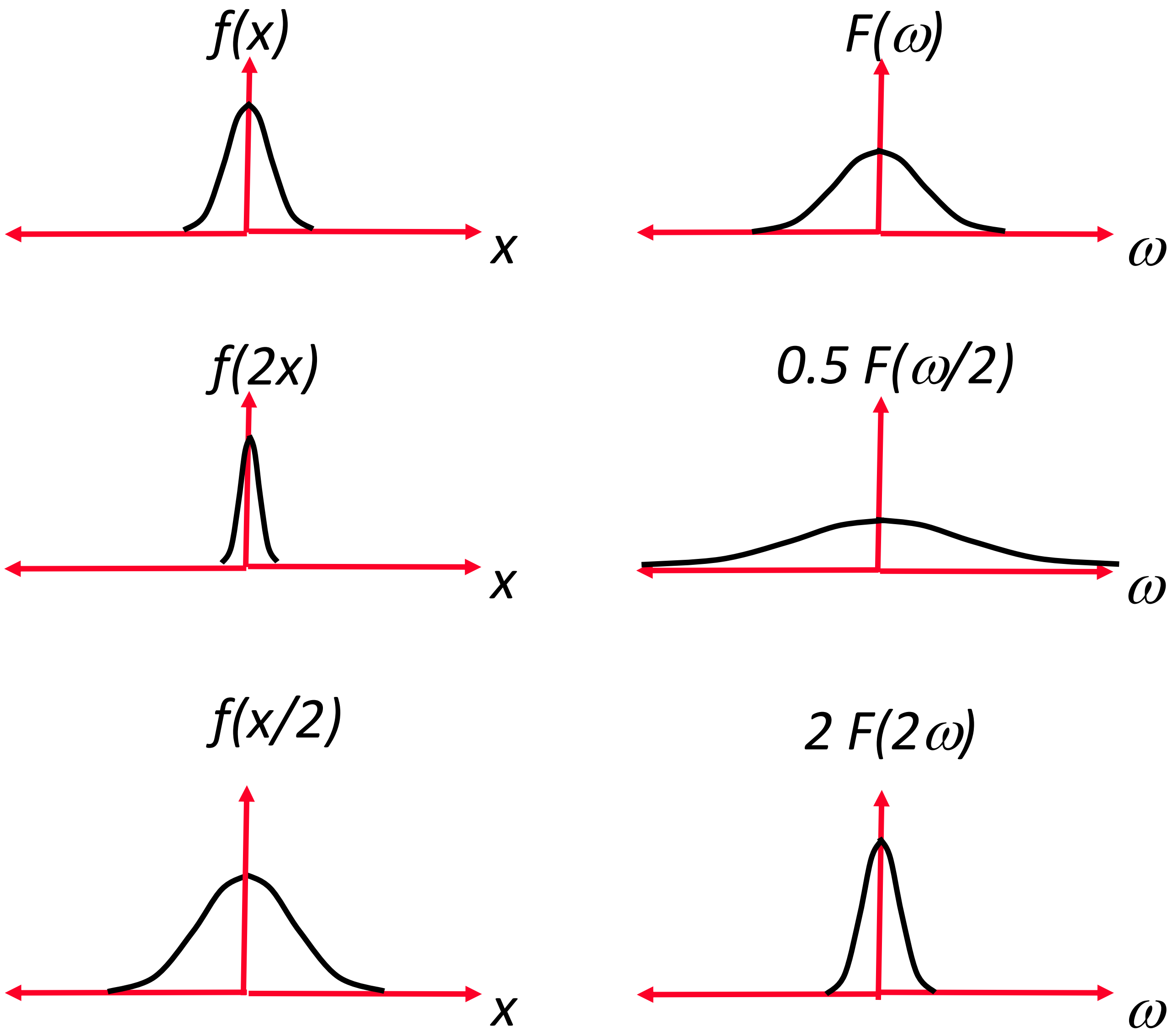
$$\Phi(f_1(x,y) + f_2(x,y)) = \Phi(f_1(x,y)) + \Phi(f_2(x,y))$$

$$\Phi(a \cdot f(x,y)) = a \cdot \Phi(f(x,y))$$

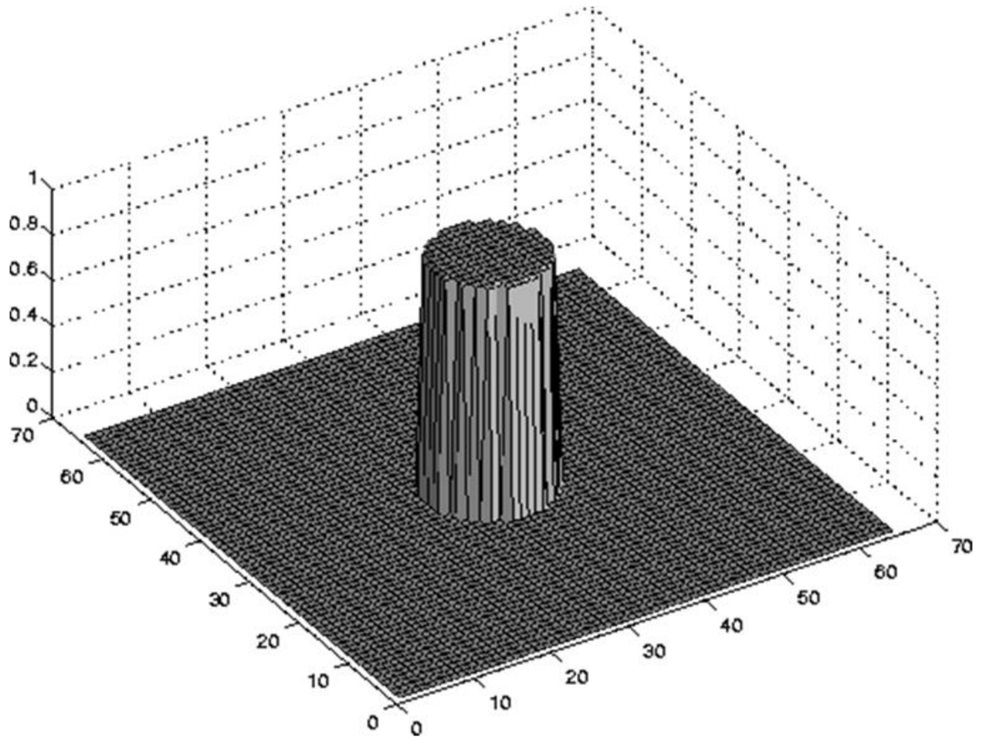
$$\Phi(f(ax,by)) = \frac{1}{|ab|} F\left({}^{u}/_{a}, {}^{v}/_{b}\right)$$

# Change Scale: Examples

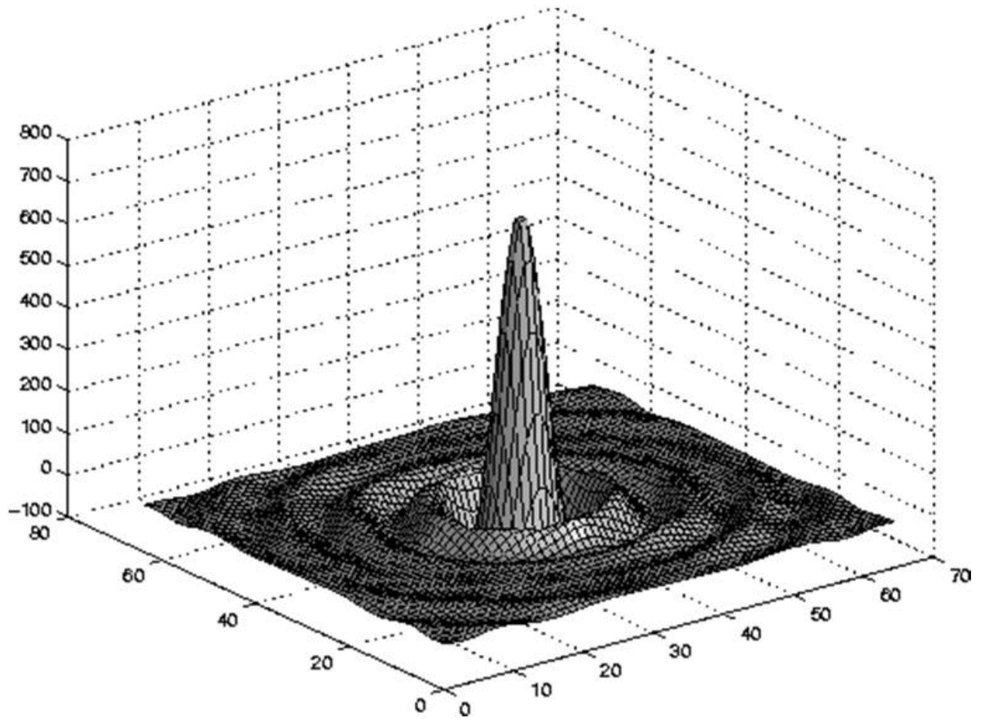
$$\Phi(f(ax,by)) = \frac{1}{|ab|}F\left({}^{u}/_{a},{}^{v}/_{b}\right)$$



# 2D Fourier




Original Function




spectrogram

# Discrete Derivatives Using Fourier

Derivatives are defined over continuous functions.

How can we compute derivative of a sequence of numbers?

One Possibility: Use Fourier Transform

$$f(x) = \boxed{\sum_u F(u) e^{\frac{2\pi i u x}{N}}}$$

Derivative – Increase high frequencies

$$F(u) \to uF(u)$$

$$f'(x) = \left( \sum_u F(u) e^{\frac{2\pi i u x}{N}} \right)' = \sum_u F(u) \left( e^{\frac{2\pi i u x}{N}} \right)' =$$

$$= \frac{2\pi i}{N} \boxed{\sum_u uF(u) e^{\frac{2\pi i u x}{N}}}$$

# 2D Derivatives using Fourier

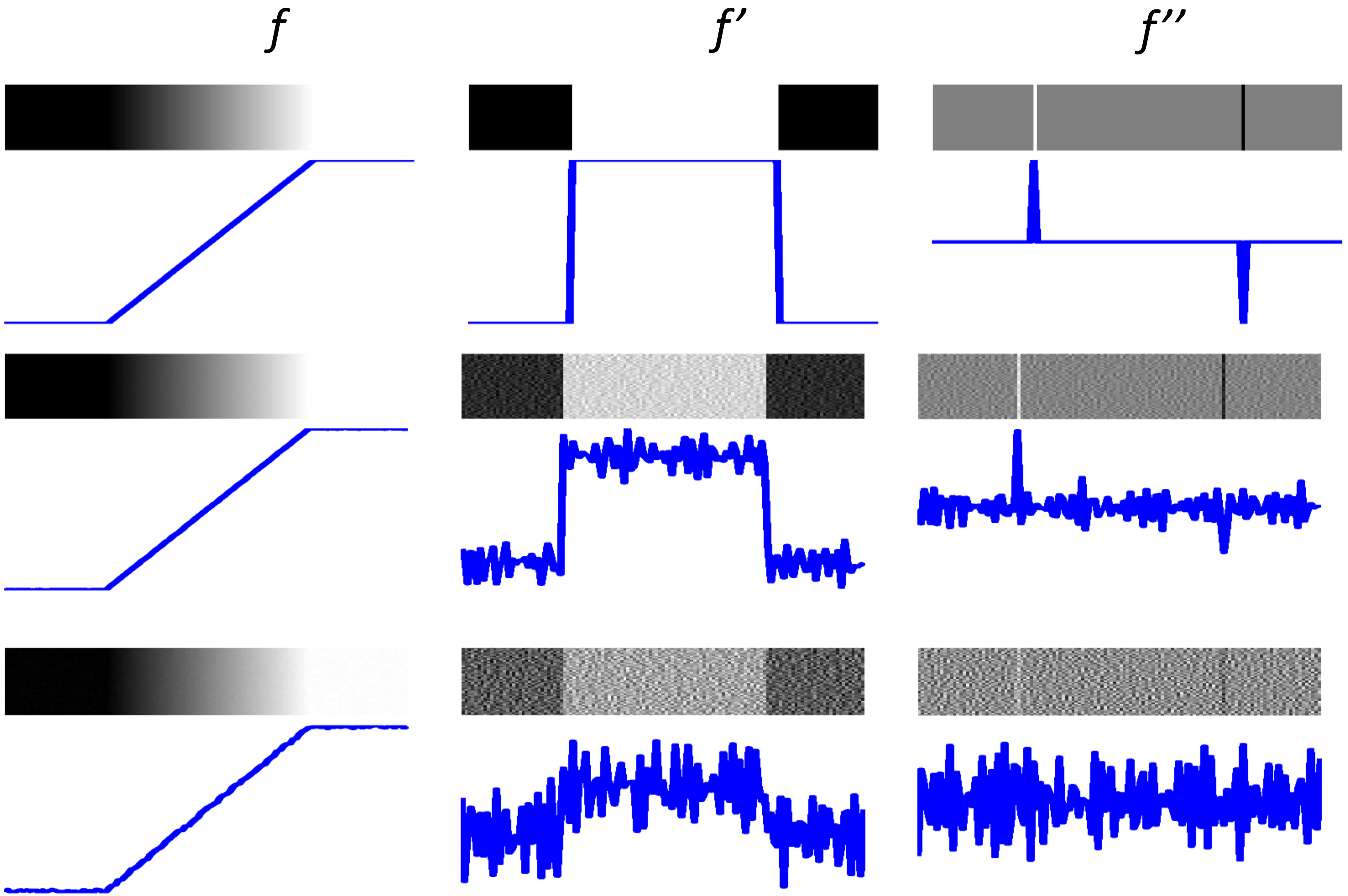
- To compute the *x* derivative of *f(x,y)* :
  1. Compute the Fourier Transform *F(u,v)*
  2. Multiply Fourier coefficient *F(u,v)* by $\frac{2\pi i}{N}u$
  3. Compute the Inverse Fourier Transform
- To compute the *y* derivative of *f(x,y)* :
  1. Compute the Fourier Transform *F(u,v)*
  2. Multiply Fourier coefficient *F(u,v)* by $\frac{2\pi i}{N}v$
  3. Compute the Inverse Fourier Transform

# Derivative as a Fourier Filter

$$f'(x) = \frac{2\pi i}{N} \sum_u uF(u) e^{\frac{2\pi i u x}{N}}$$

- Multiply Fourier Coefficient $F(u)$ with $\frac{2\pi i}{N} u$
- Amplifies higher frequencies (and Noise)
  - **Since Noise** has more high frequency than normal images
  - Derivatives amplify noise
- Cancels DC ($F(0)$, the image average, becomes zero)

# Effect of Noise on Derivatives

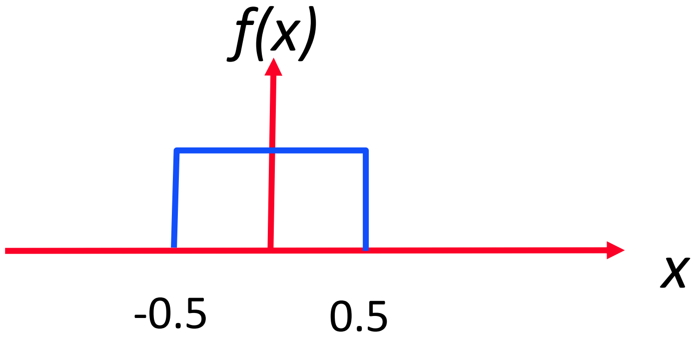


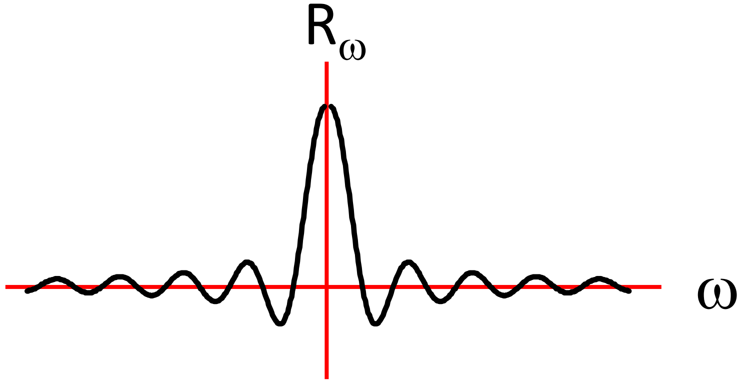

Note: This slide neglects the cyclic effect, and derivatives here do not sum to 0.

# Fourier of Special Functions

## The Window (Box) Function (Rect):

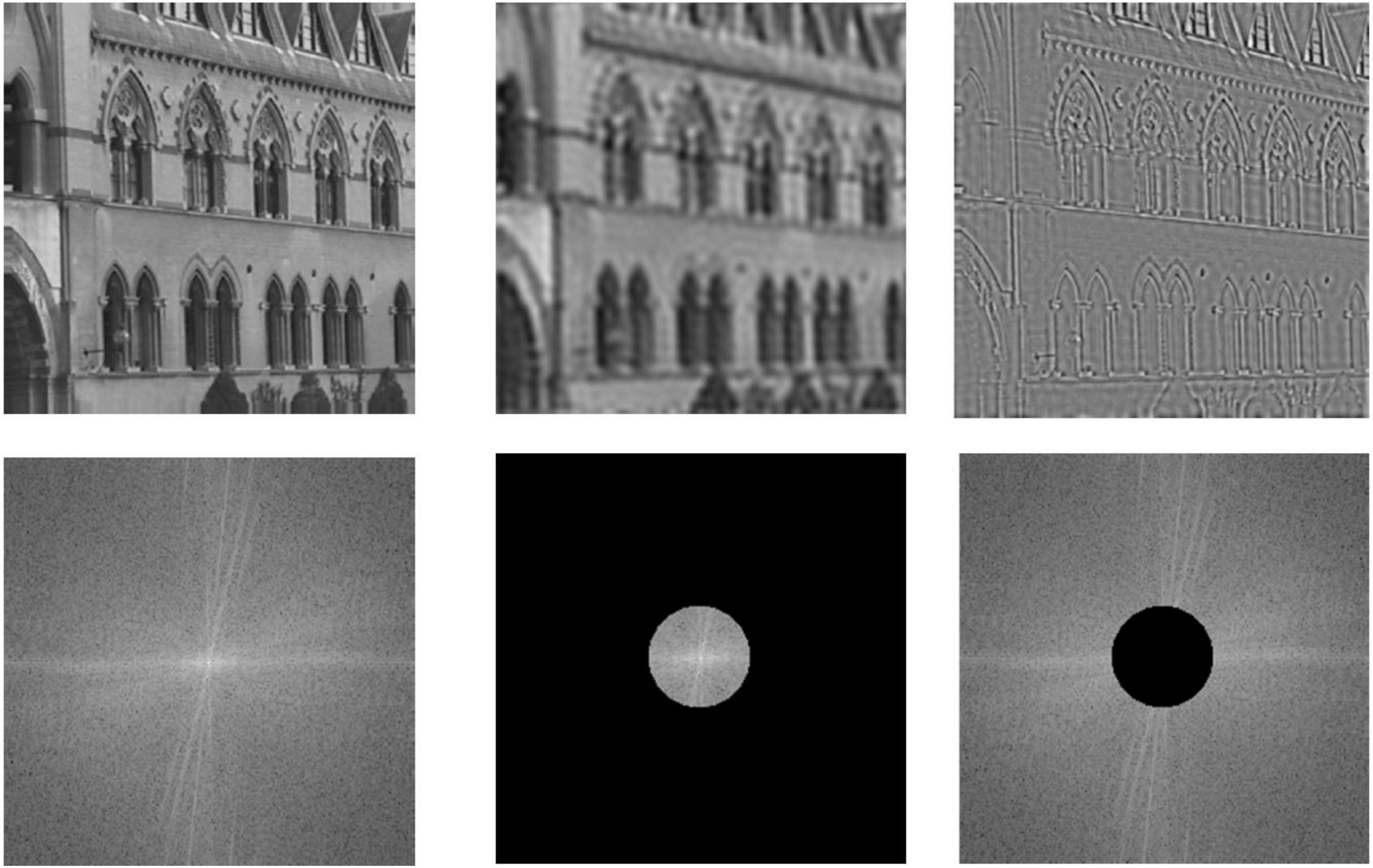
$$\text{rect}(x) = \begin{cases} 1 & \text{if}|x| < \dfrac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$



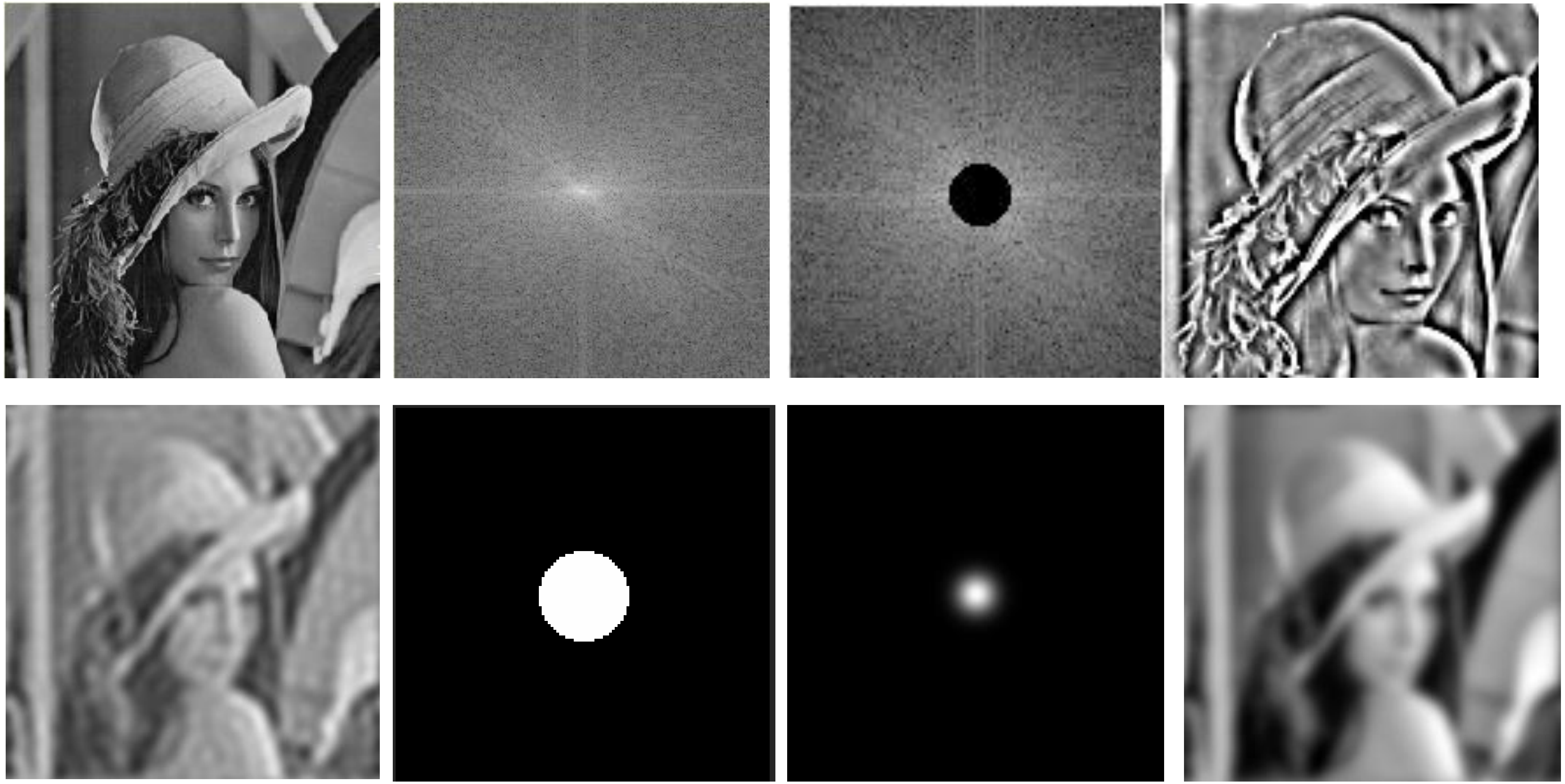

$$F(\omega) = \frac{\sin(\pi\omega)}{\pi\omega} = \text{sinc}(\pi\omega)$$

original | low pass | high pass



# Compare Filters

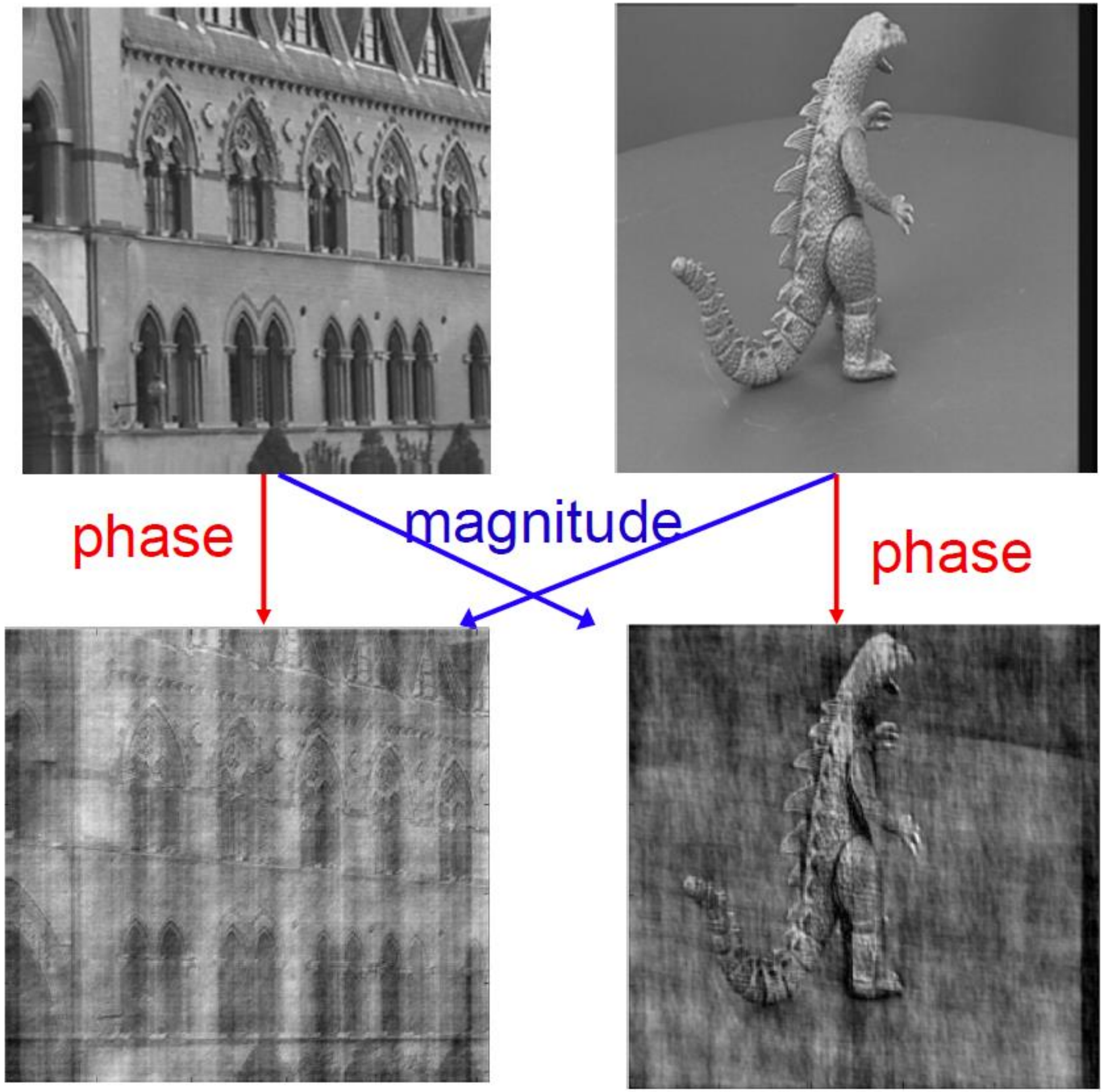
Original　　Fourier　　Hi-Pass Box Filter



Low Pass
Box Filter

Low Pass
Gaussian Filter

# Effects of Phase and Magnitude



# Fourier Transform Pairs

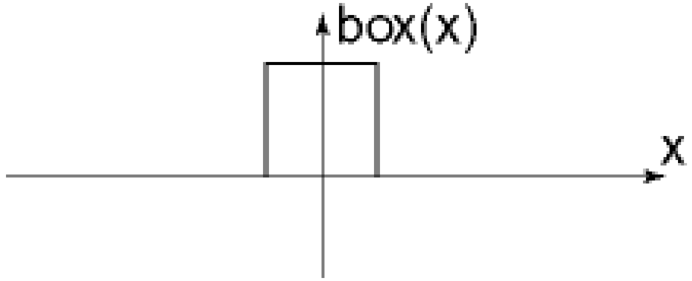
$$f(x)$$

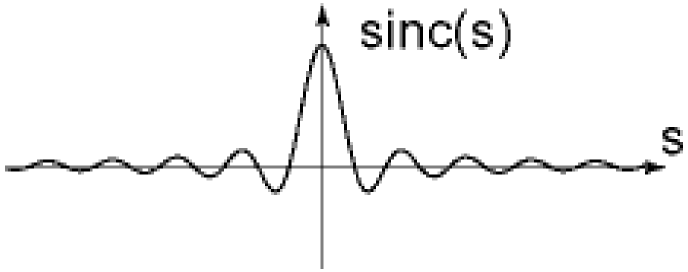
$$F(s) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi s x} dx$$

Spatial domain

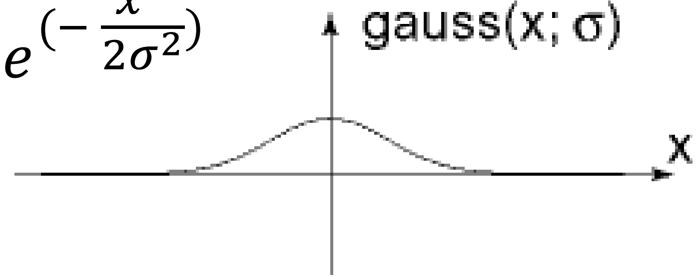
Frequency domain

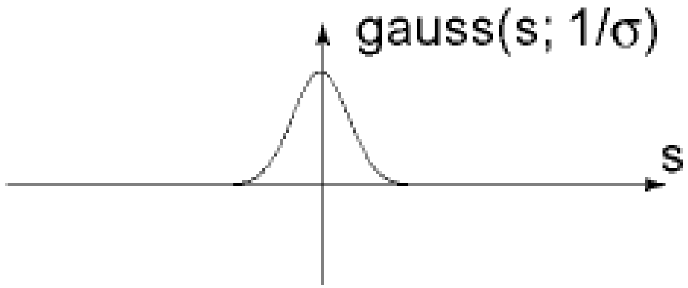
$$\text{sinc}(x) = \frac{\sin(x)}{x}$$

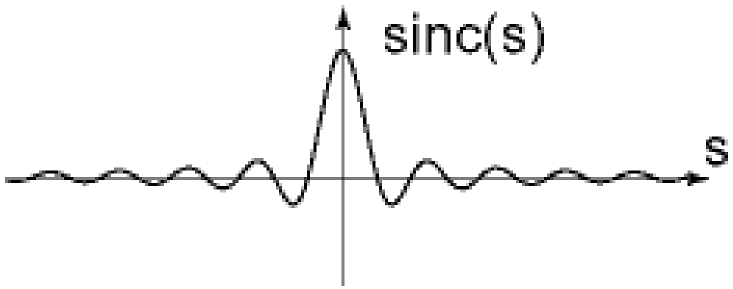


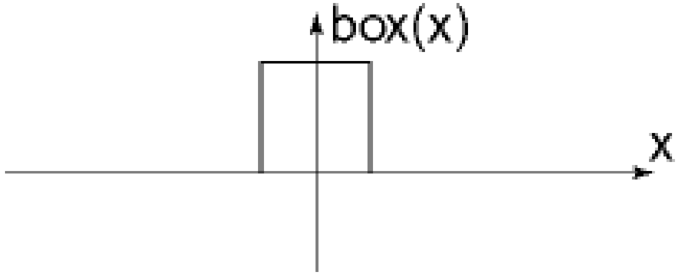




$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{x^2}{2\sigma^2}\right)}$$



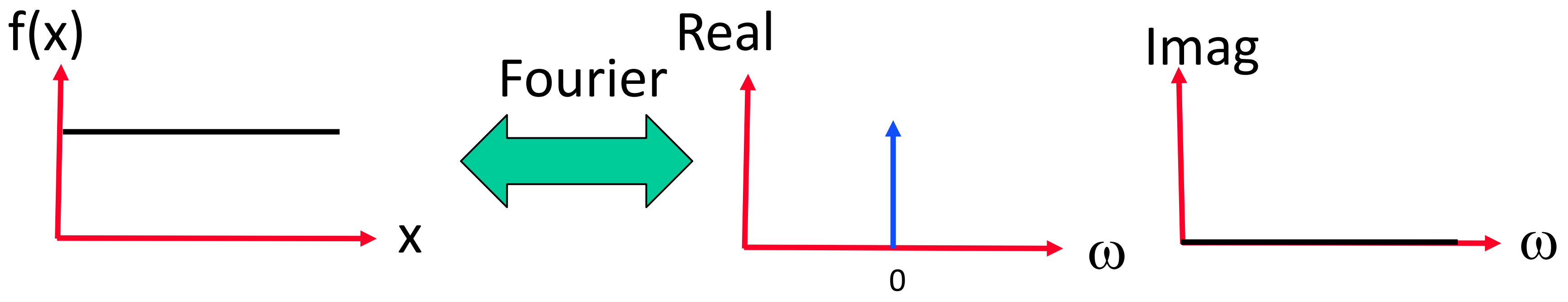



# Fourier of Special Functions

## The Constant Function:

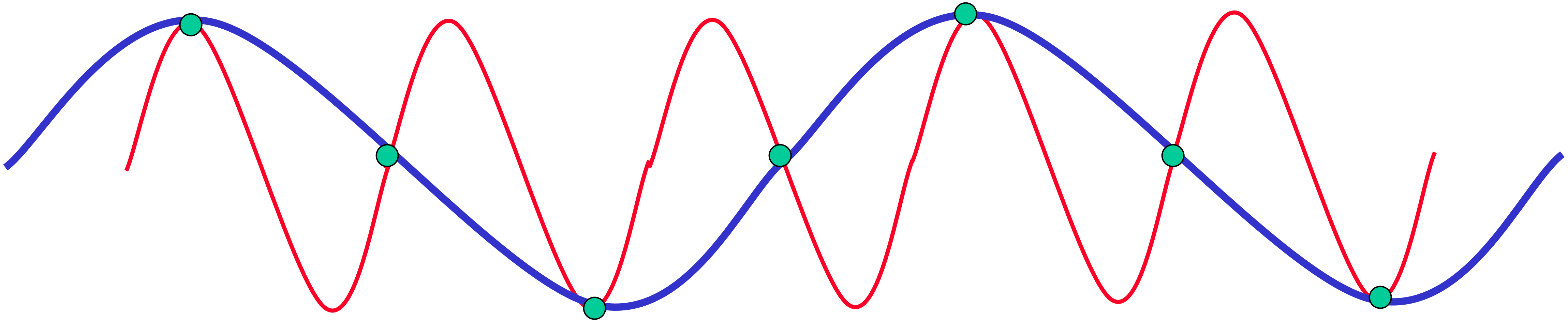
$$f(x) = 1$$

$$F(u) = \tfrac{1}{N}\sum_{x=0}^{N-1} 1\, e^{\frac{-2\pi i u x}{N}}$$



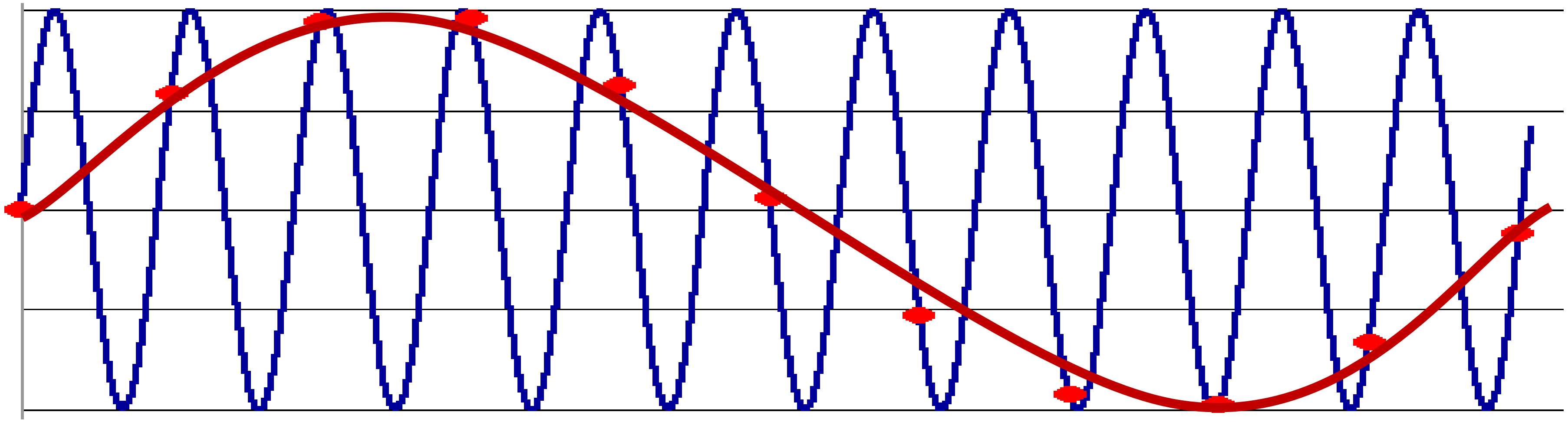
# Aliasing

- Sampling can result in aliasing.
- Example: Sampling at 1.5π



- To avoid aliasing, distance between sampling points should be <u>less than </u>½ of wavelength (Niquest…)

# Aliasing



# Sampling

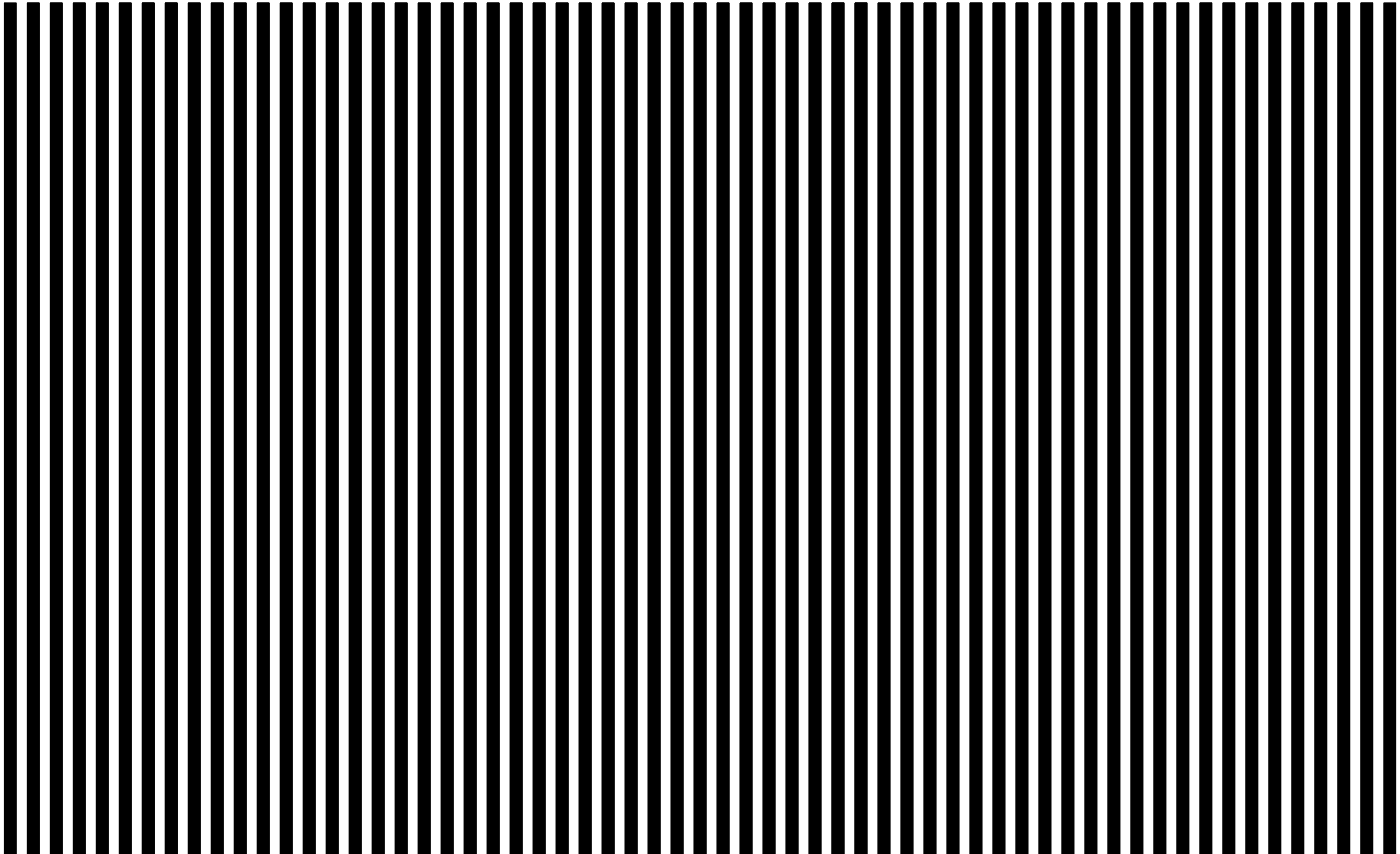
## *The most important slide in course!*

- **Blur before you sample** (Low-pass filter: reduce the highest frequencies)
- Sampling without low-pass results in aliasing.

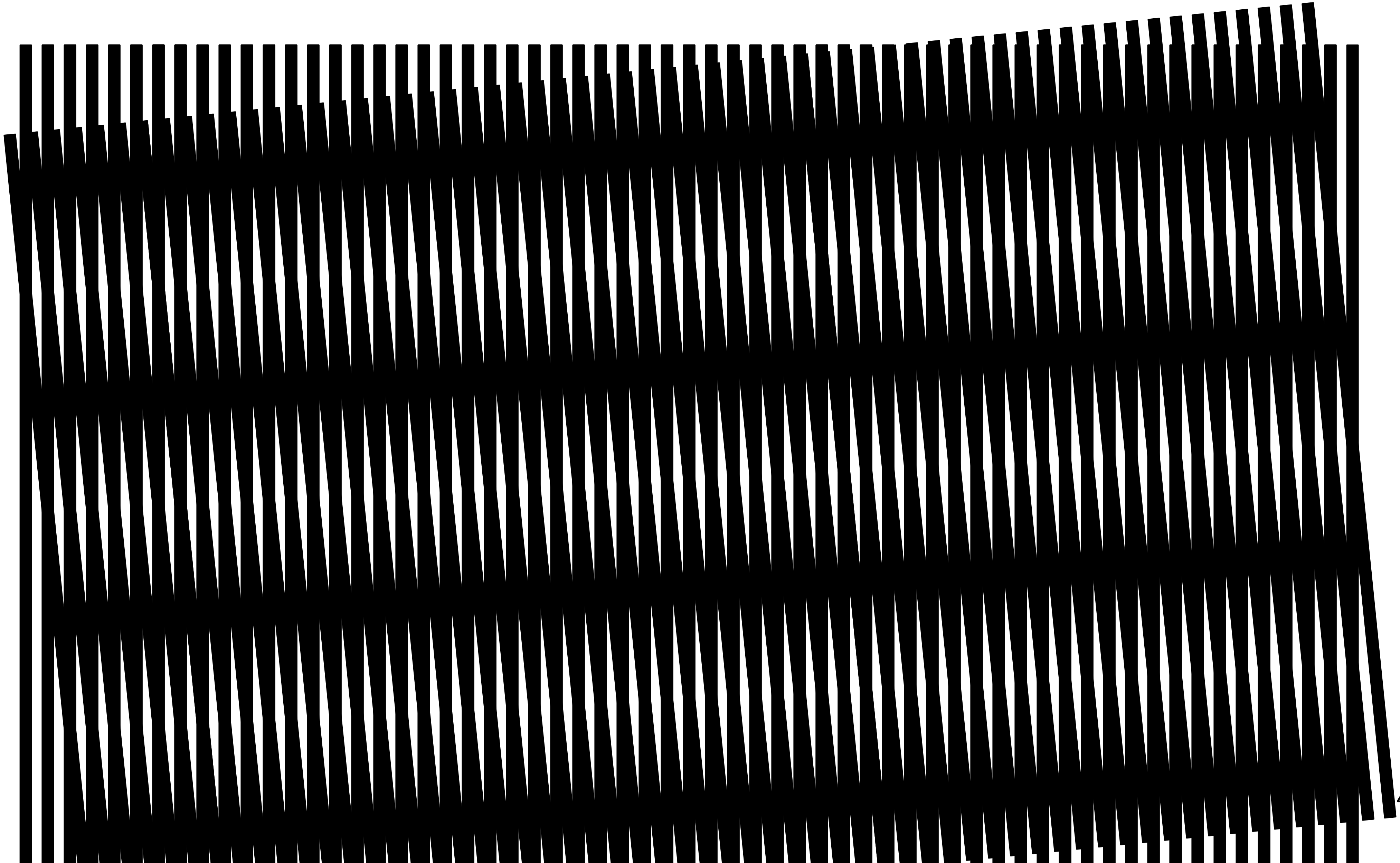
- How NOT to shrink an image:
  - sample every other pixel

- Blur before you sample!

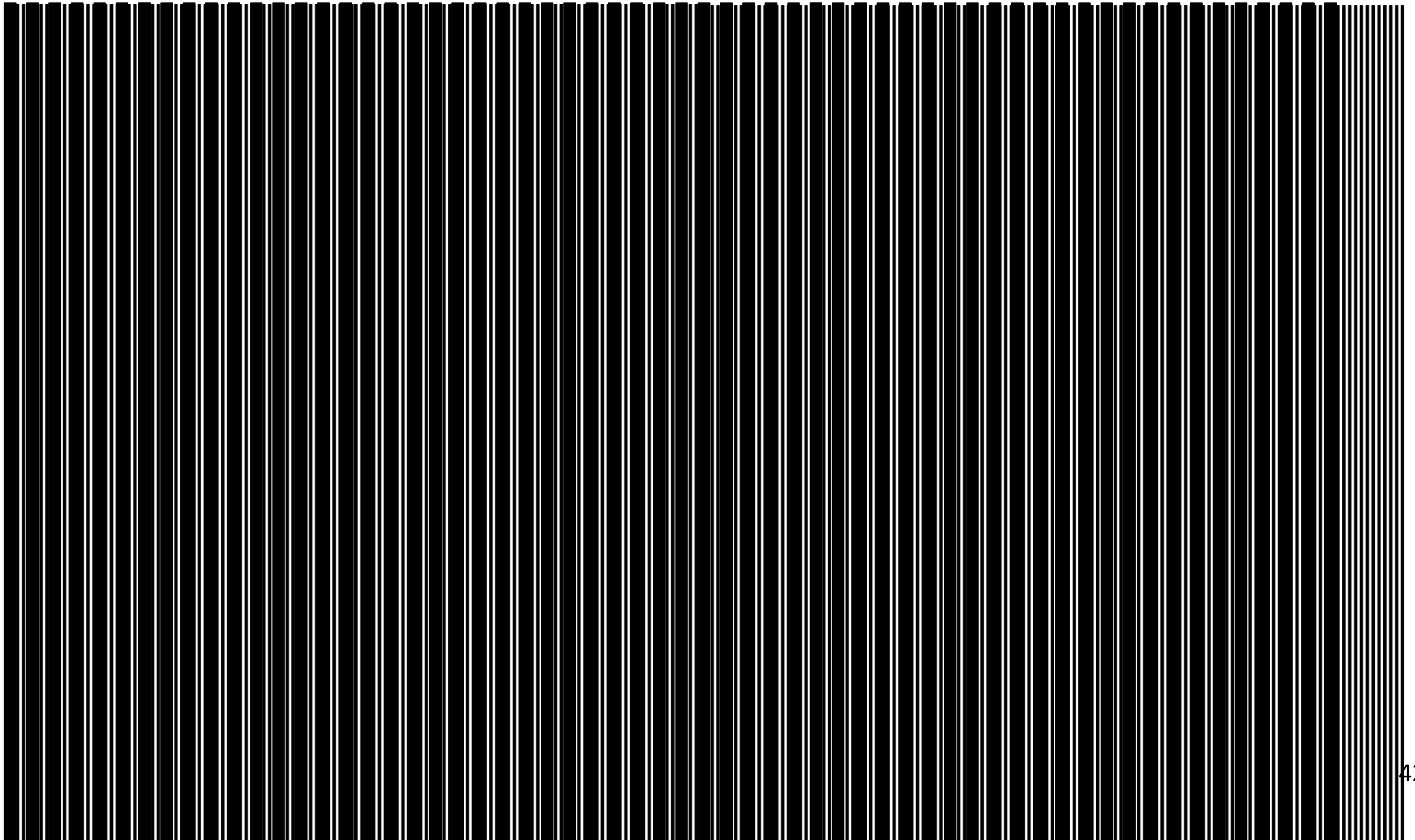
# Image Aliasing Example



# Image Aliasing Example – 1 sample/cycle



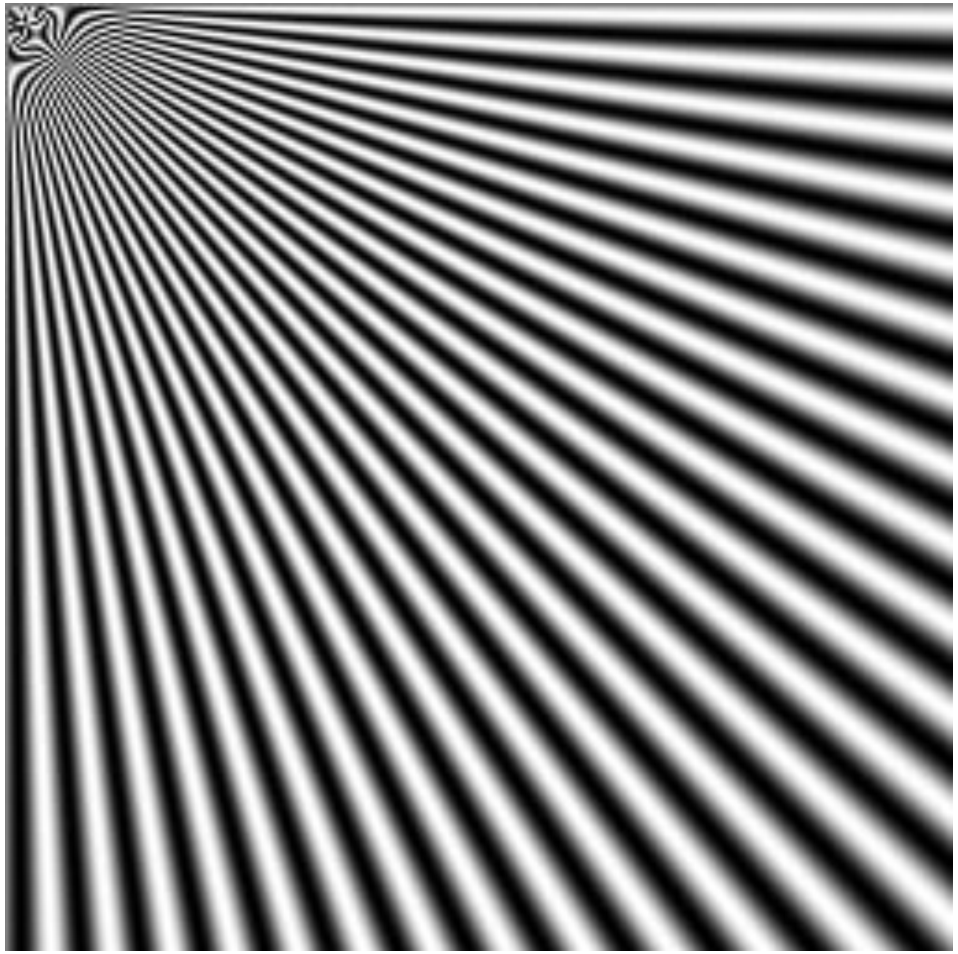
# Image Aliasing Example – 3 samples/cycle



# Image Aliasing Example


- Assume a column is 1 pixel wide. B/W cycle length is 2. Good sampling must be at distances <1.
- Sampling every second column will give either a solid black or a solid white.
- Blur before sample will give a solid gray regardless of shift.

# Aliasing Example



# Aliasing Example

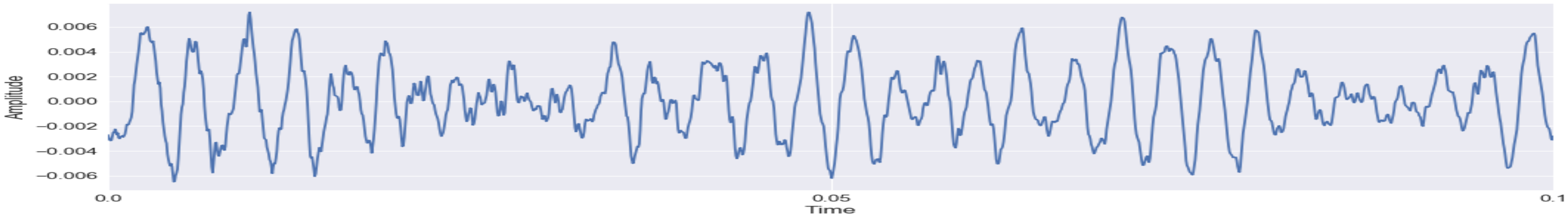
# Signal Resizing (Example: *N samples → N/2 samples*)

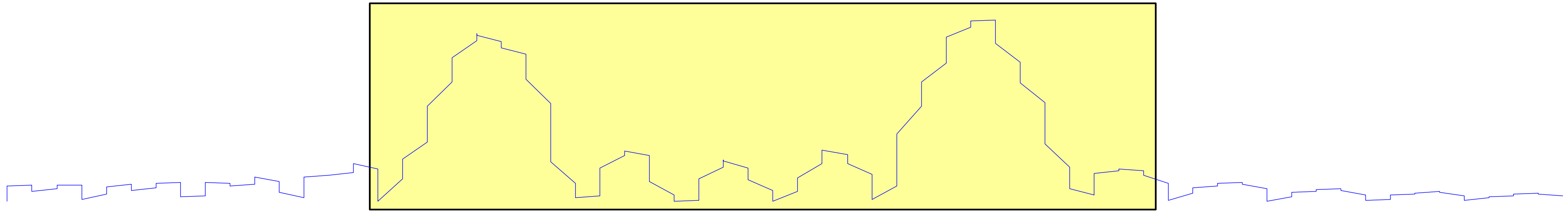
Use Fourier

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\, e^{\frac{-2\pi i u x}{N}}$$

$$f(x) = \sum_{u=-N/4}^{N/4-1} F(u)\, e^{\frac{2\pi i u x}{N/2}}$$

1. Compute Fourier (*N* samples to *N* coefficients)
2. Bring u=0 to center (FFT Shift)
3. Crop Fourier from *N* to *N/2 coefficients - remove N/2 high frequencies*
4. Compute Inverse Fourier (*N/2* coefficients to *N/2* Samples)

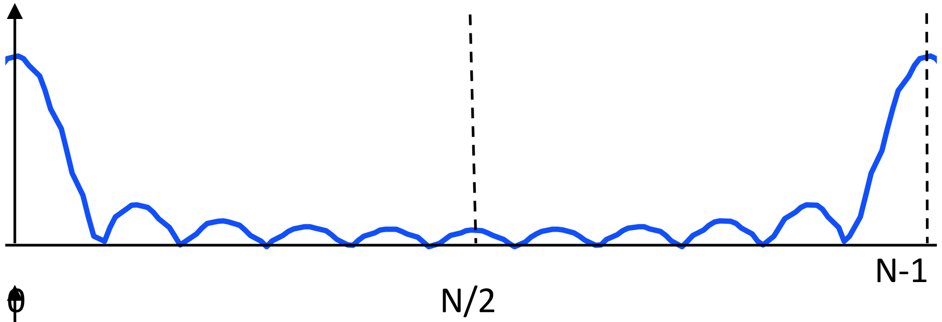


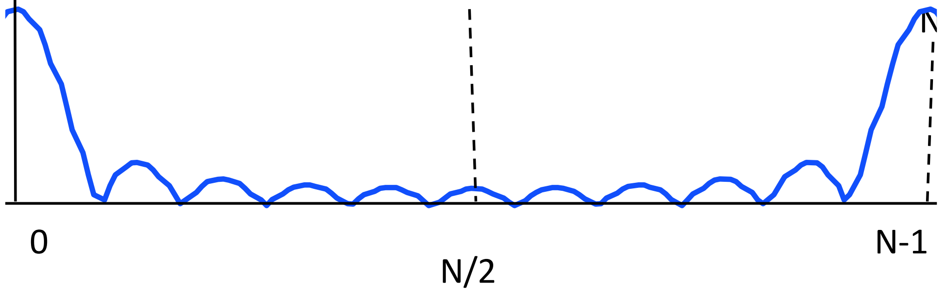
u=-N/2 u=0 u=N/2

# Resizing (1D)

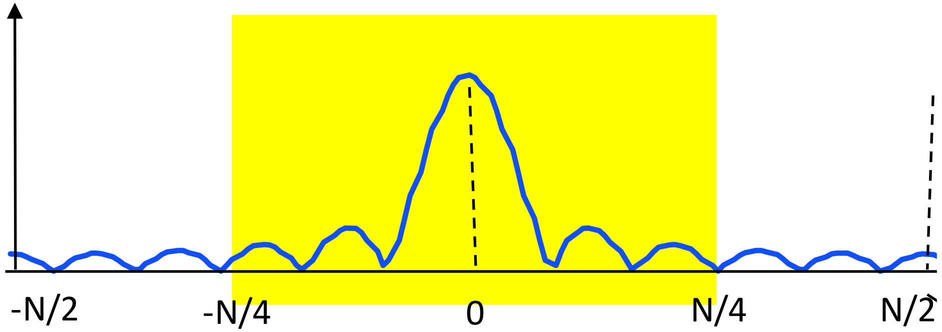
FFT of Original Signal

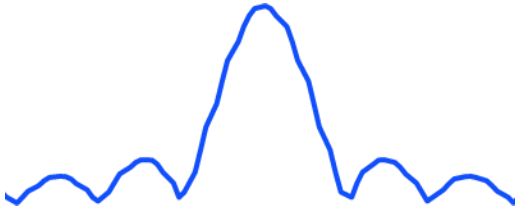



fftshift




|F(u)| Clip



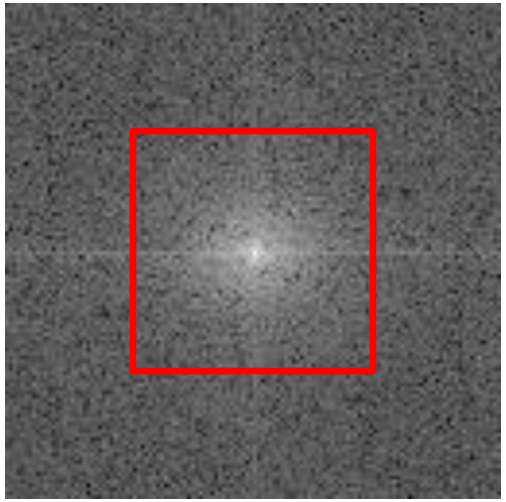



Inverse FFT

Shorter Signal

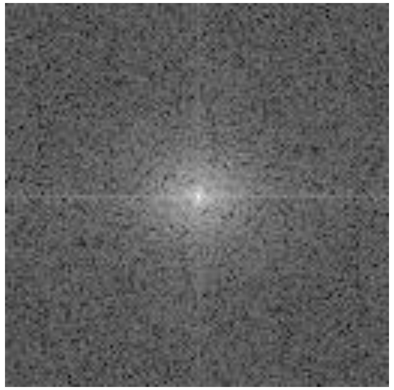
# Image Reduction (E.g. $N \times N \rightarrow N/2 \times N/2$)

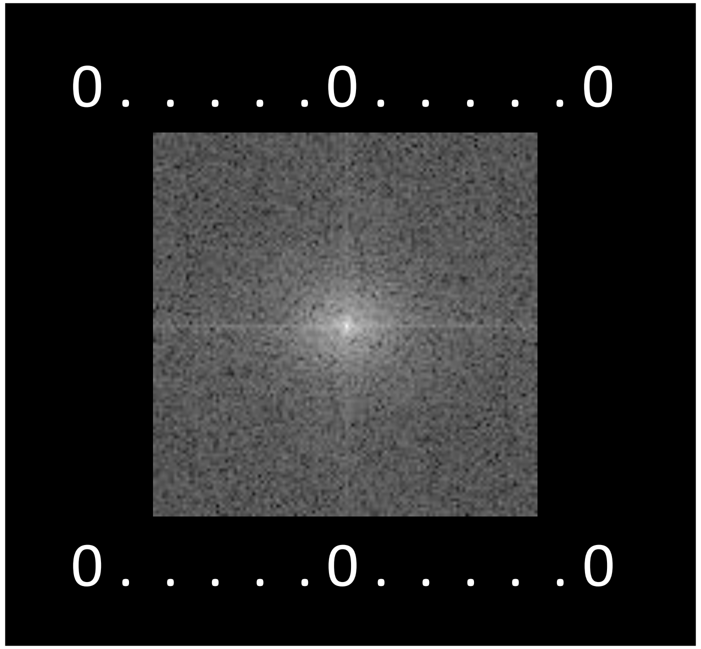
1. **Blur** and  Subsample every 2nd pixel in every 2nd row
2. Use Fourier (Can resize to any new size)
    1. Compute Fourier ($N \times N$) , Shift (0,0) to center
    2. Crop Fourier (e.g. $N/2 \times N/2$): Ideal low pass
    3. Compute Inverse Fourier

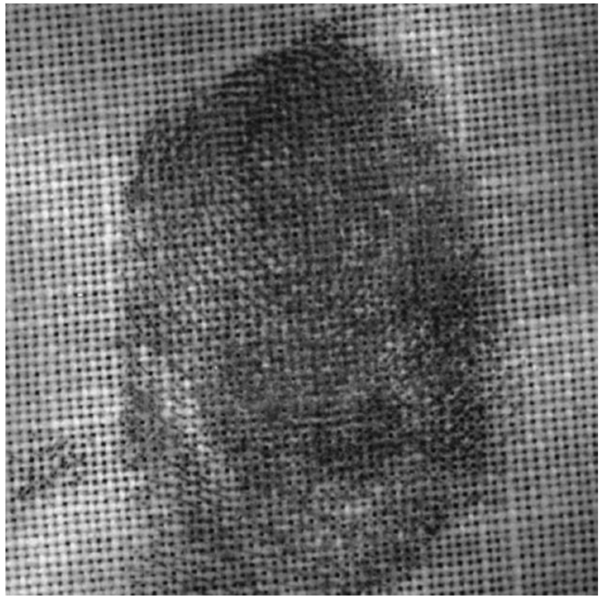
# Image Expansion (E.g. $N/2 \times N/2 \rightarrow N \times N$)

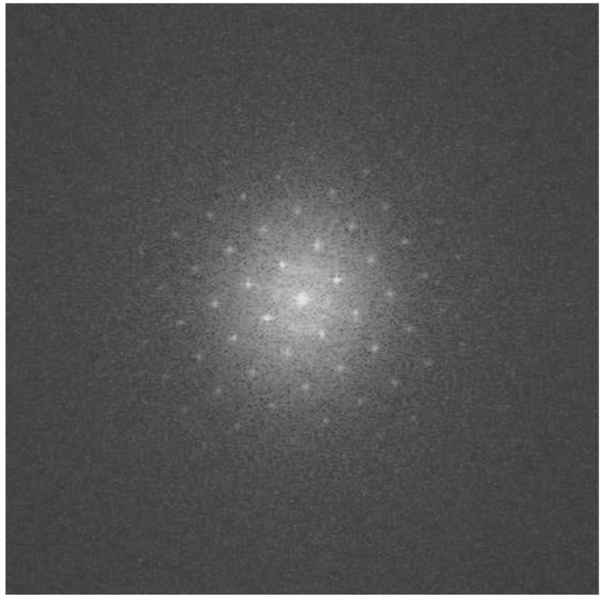
1. Compute Fourier ($N \times N$)
2. Pad Fourier with zeros (E.g. $2N \times 2N$)
3. Compute Inverse Fourier ($2N \times 2$N)



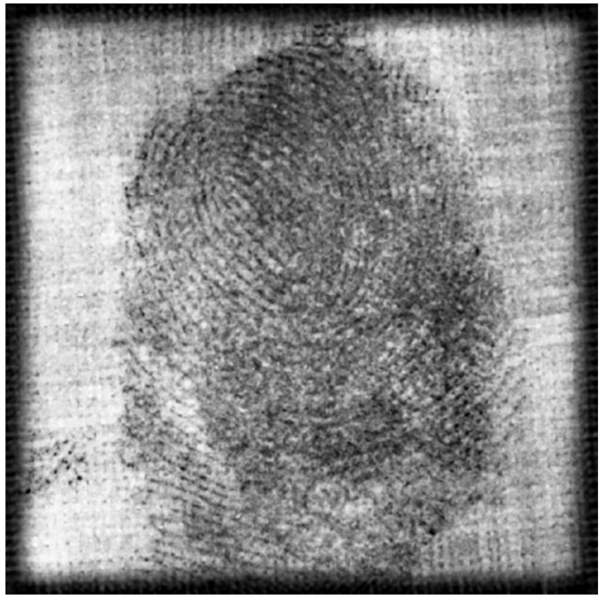


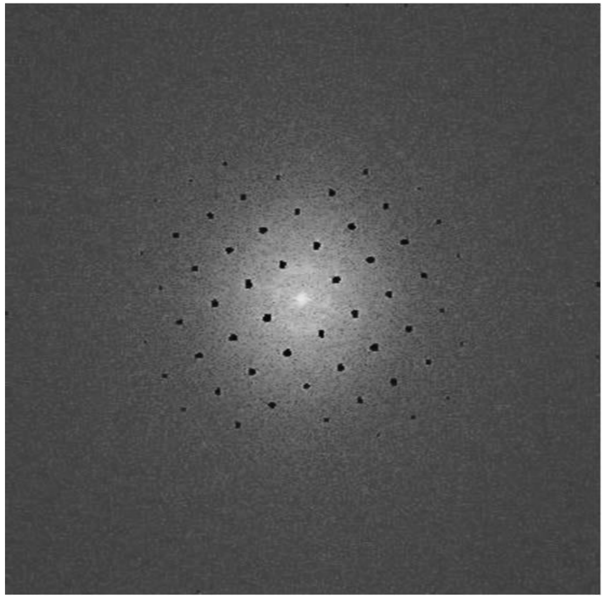

# Enhancement – Remove periodic background





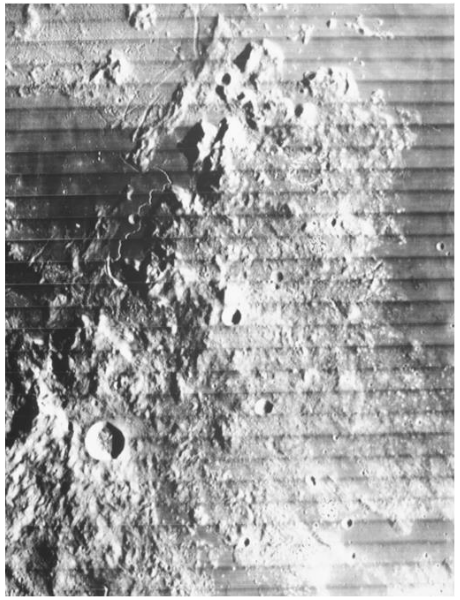
$|F(u,v)|$

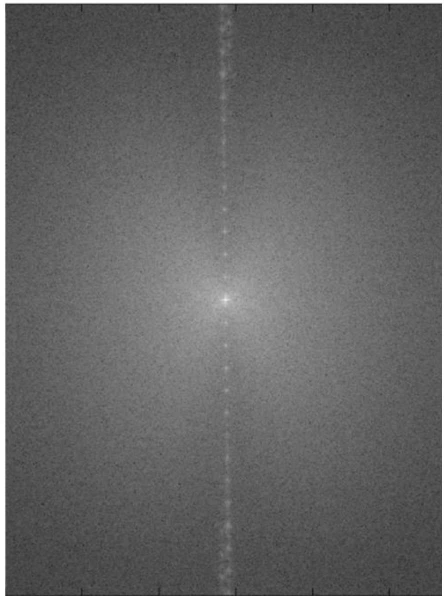




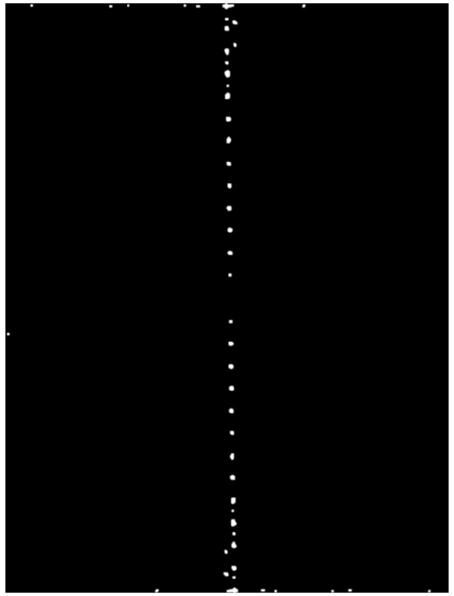
Fourier with peaks removed

# Enhancement – Remove periodic noise





$|F(u,v)|$



Filter to remove peaks