Lecture 3: Signal processing

Andrew Owens

- PS1 due next Tuesday
- Updated holiday office hours:

- Questions?

Today

- A few more filters
- Fourier analysis

Laplacian filter

- Used to detect object boundaries and other salient image structures
- The Laplacian operator is defined as the sum of the second order derivatives of a function:

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

- Sensitive to noise. Blur first!

Laplacian filter

- The most popular approximation is the five-point formula which consists in convolving the image with the kernel

$$\nabla_5^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Based on the approximation of the 2nd derivative filter: [1, -2, 1]

$$(f(n-1) - f(n)) - (f(n) - f(n+1)) = f(n-1) - 2f(n) + f(n+1)$$

Laplacian filter

- The most popular approximation is the five-point formula which consists in convolving the image with the kernel

$$\nabla_5^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

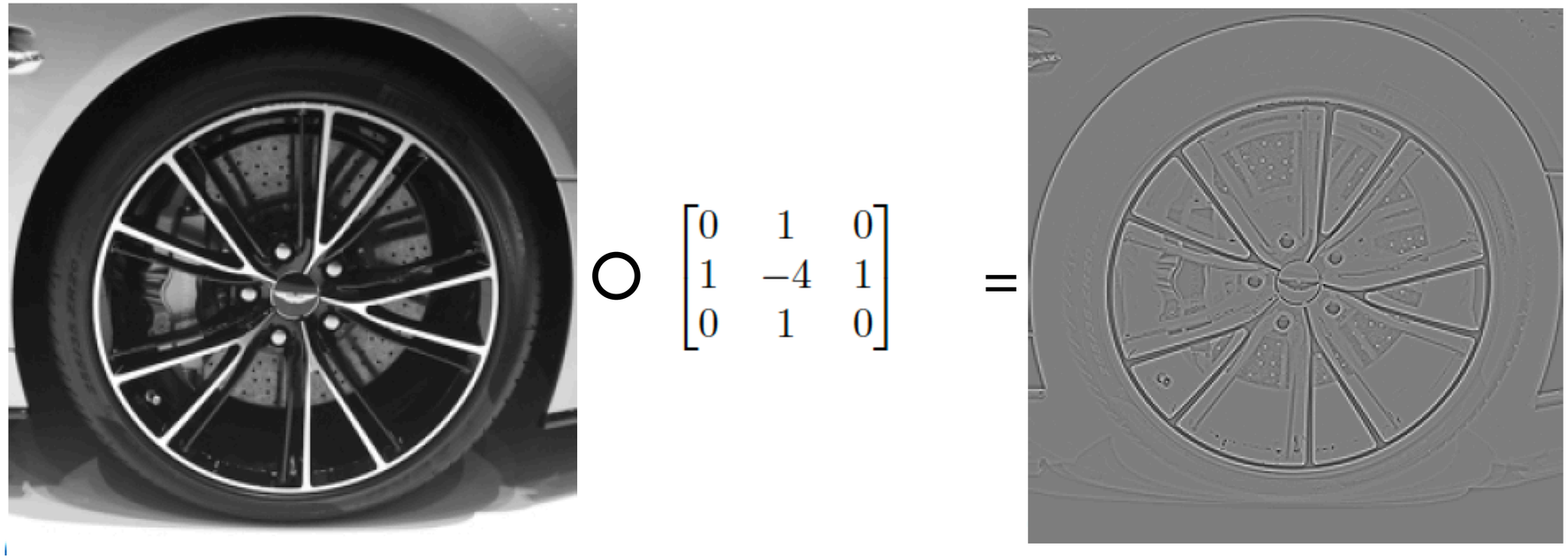
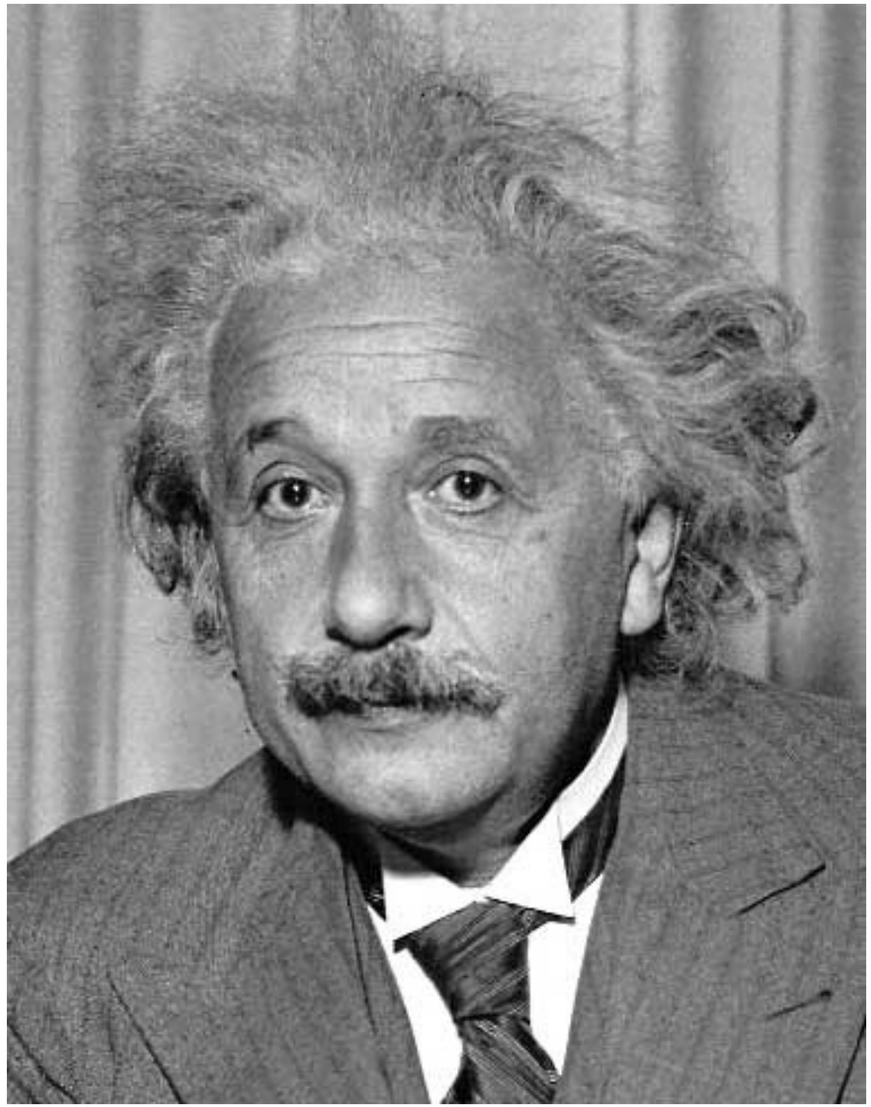


Image patches as filters

- Goal: find  in an image
- What's a good similarity/distance measure between two patches?



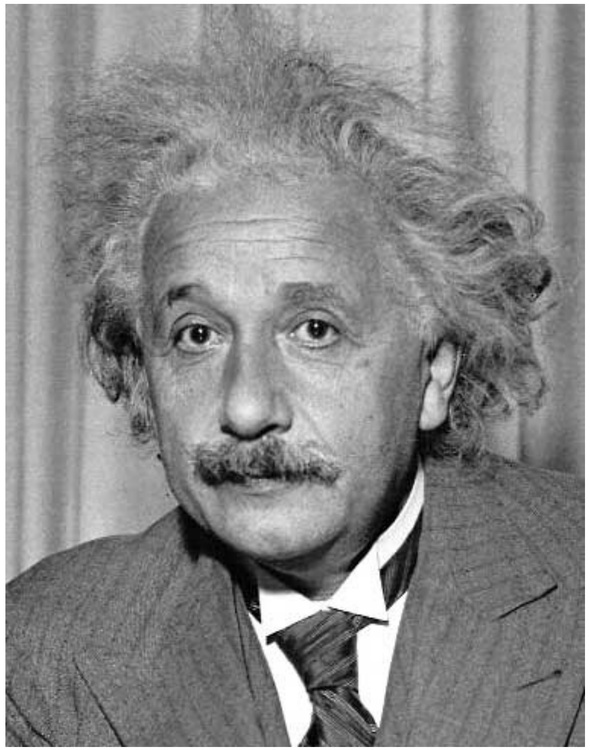
Source: D. Hoiem

Matching with filters

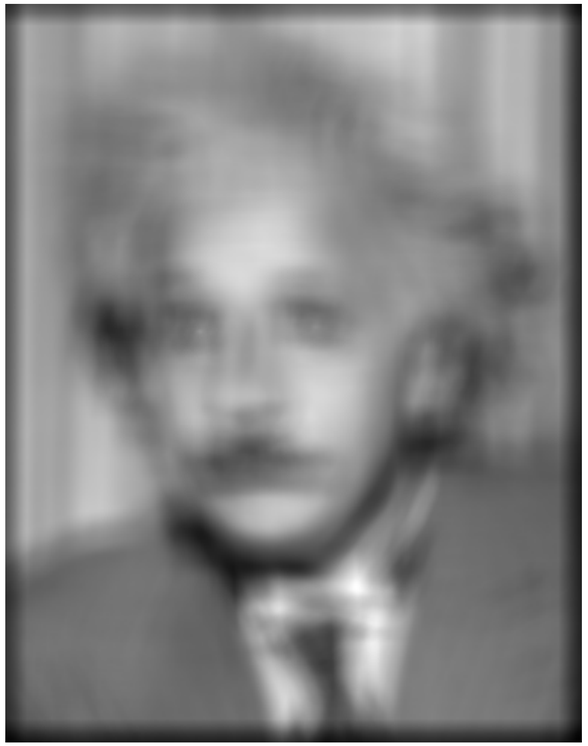
Method 1: Filter the image with 

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

f = image
g = filter



Input



Filtered Image

What went wrong?

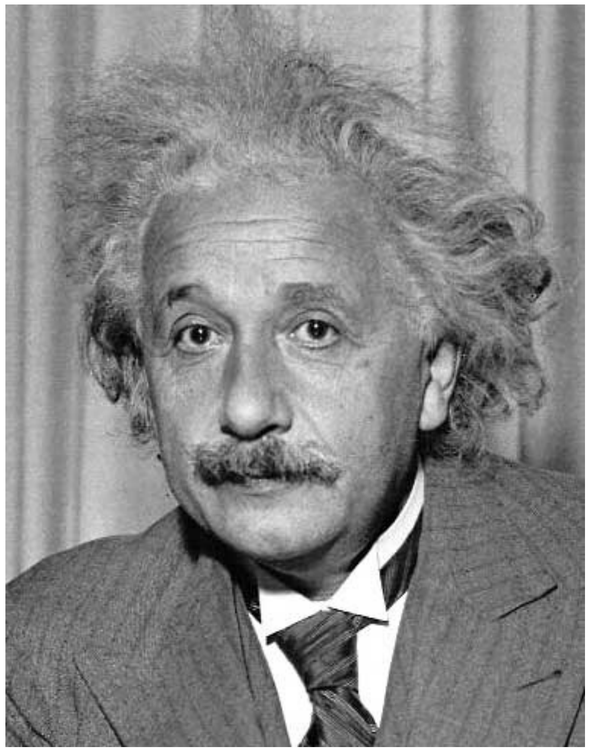
Source: D. Hoiem

Matching with filters

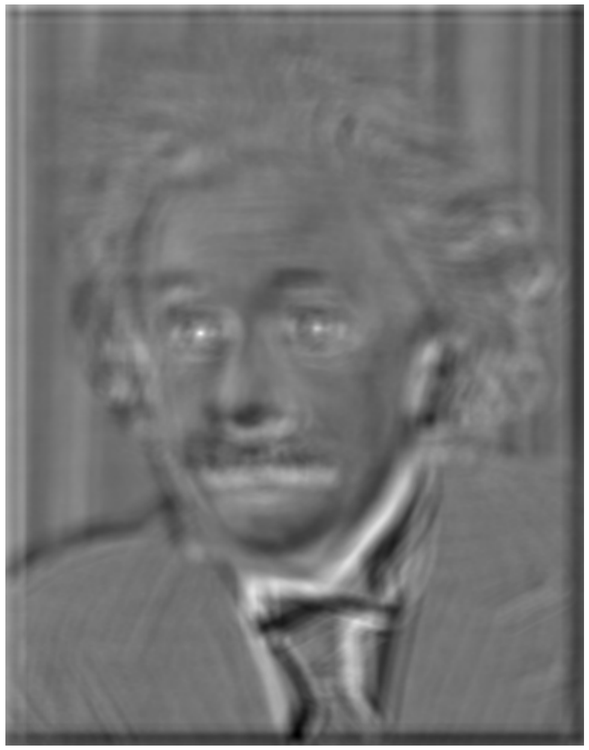
Method 2: Filter the image with zero-mean eye.

$$h[m,n] = \sum_{k,l} (f[k,l] - \bar{f})\,(g[m+k, n+l])$$

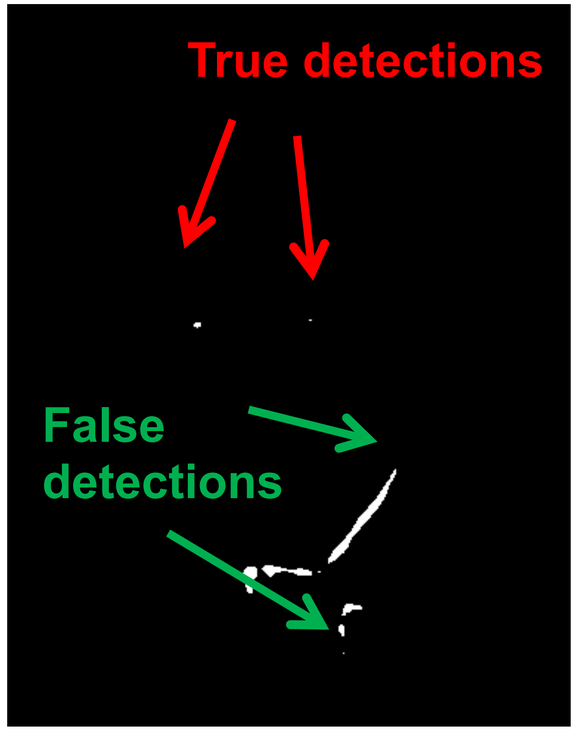
mean of f



Input



Filtered Image (scaled)



Thresholded Image

Source: D. Hoiem

Matching with filters

Method 3: Normalized cross-correlation.

mean template

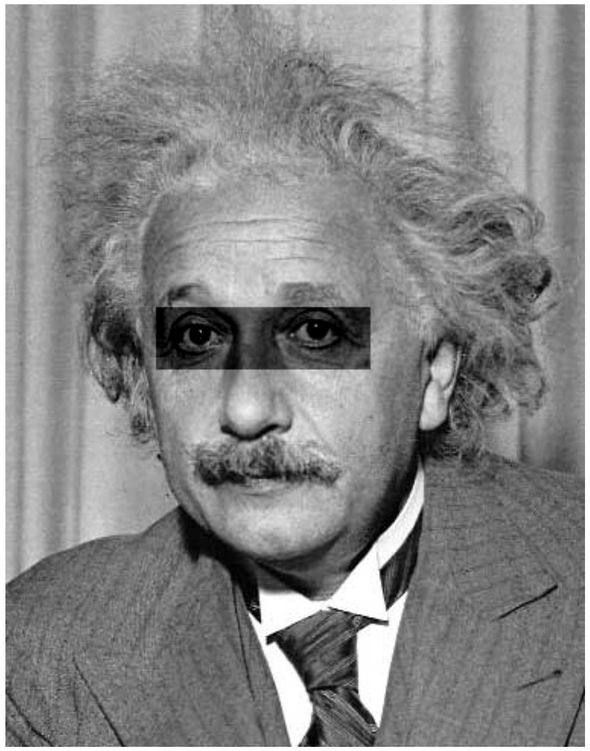
mean image patch

$$h[m,n] = \frac{\sum_{k,l}(g[k,l]-\bar{g})(f[m+k,n+l]-\bar{f}_{m,n})}{\left(\sum_{k,l}(g[k,l]-\bar{g})^2 \sum_{k,l}(f[m+k,n+l]-\bar{f}_{m,n})^2\right)^{0.5}}$$

Source: D. Hoiem

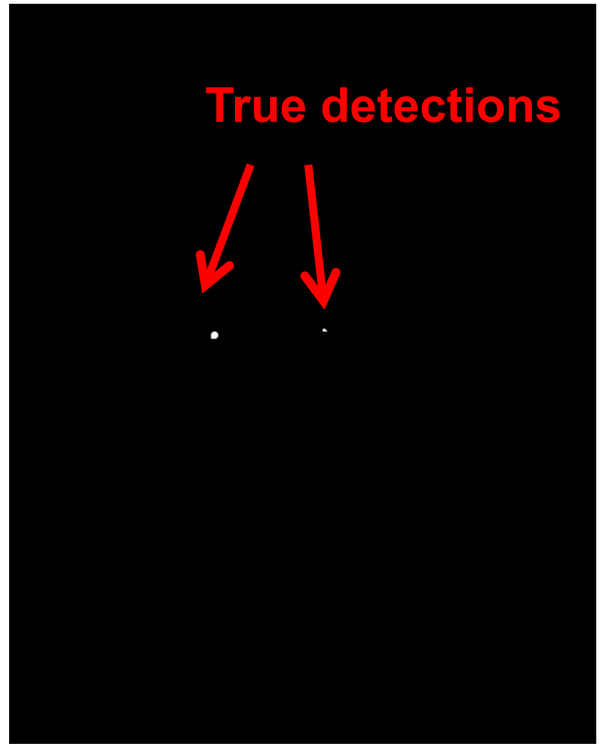
Matching with filters

Method 3: Normalized cross-correlation.



Input

Normalized X-Correlation

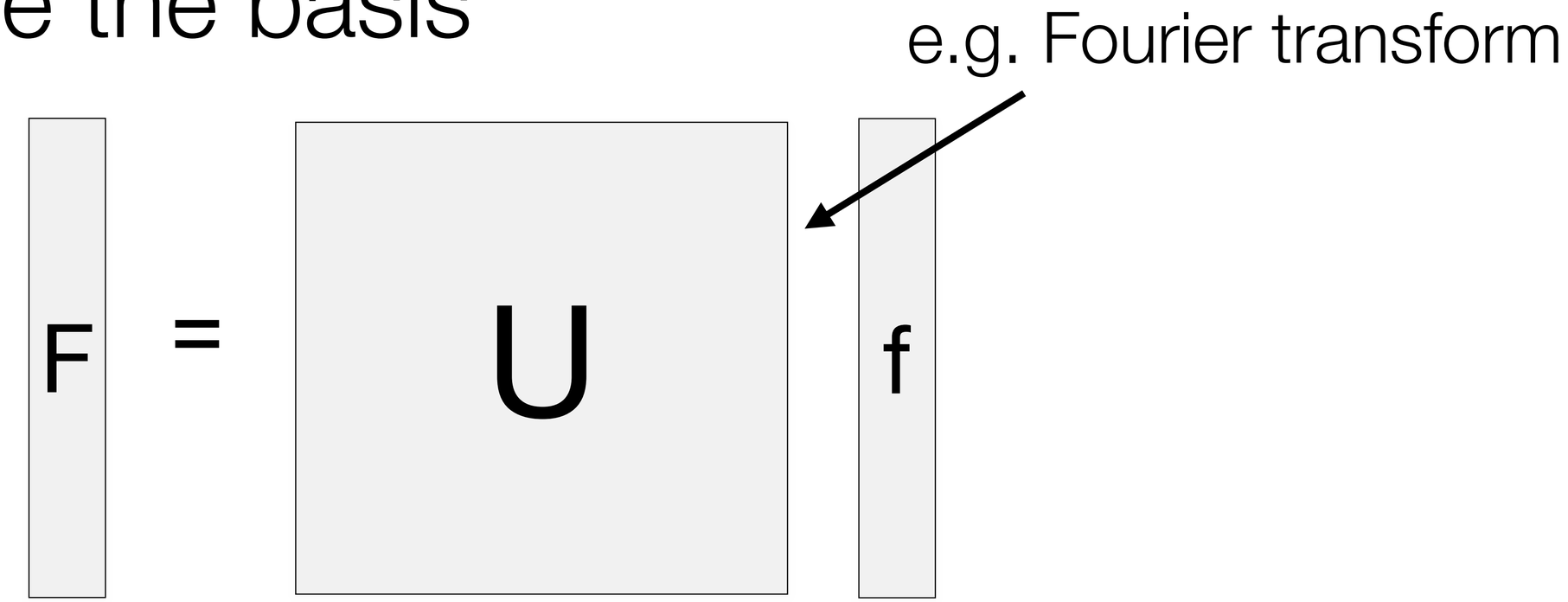


Thresholded Image

Source: D. Hoiem

Other linear transformations

- So far, we've focused on **convolution**
- What about other linear transformations?
- Change the basis

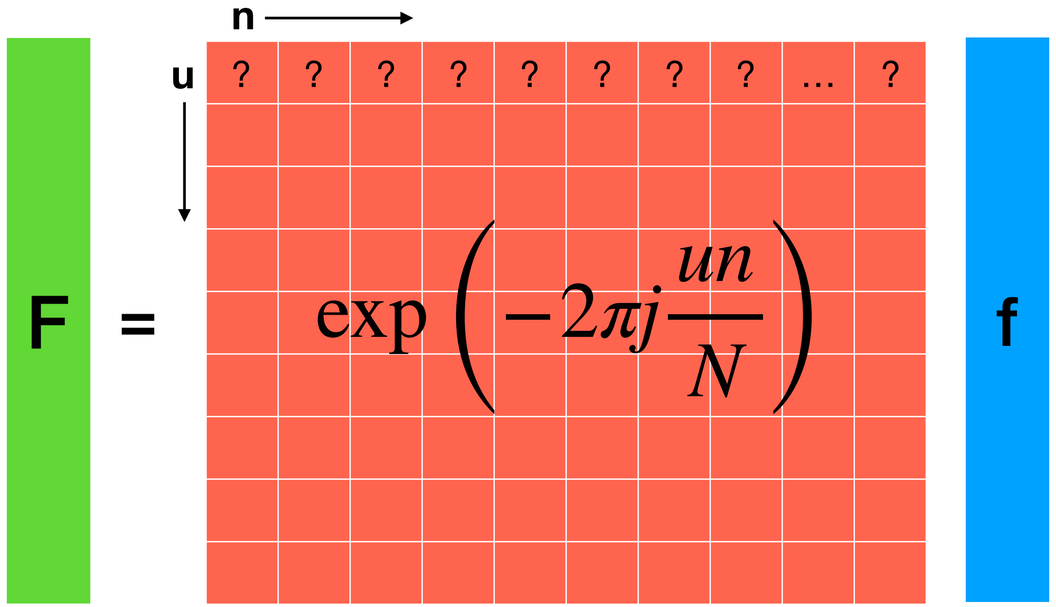


The Discrete Fourier transform

Discrete Fourier Transform (DFT) transforms a signal *f[n]* into *F[u]* as:

$$F[u] = \sum_{n=0}^{N-1} f[n] \exp\left(-2\pi j \frac{un}{N}\right)$$

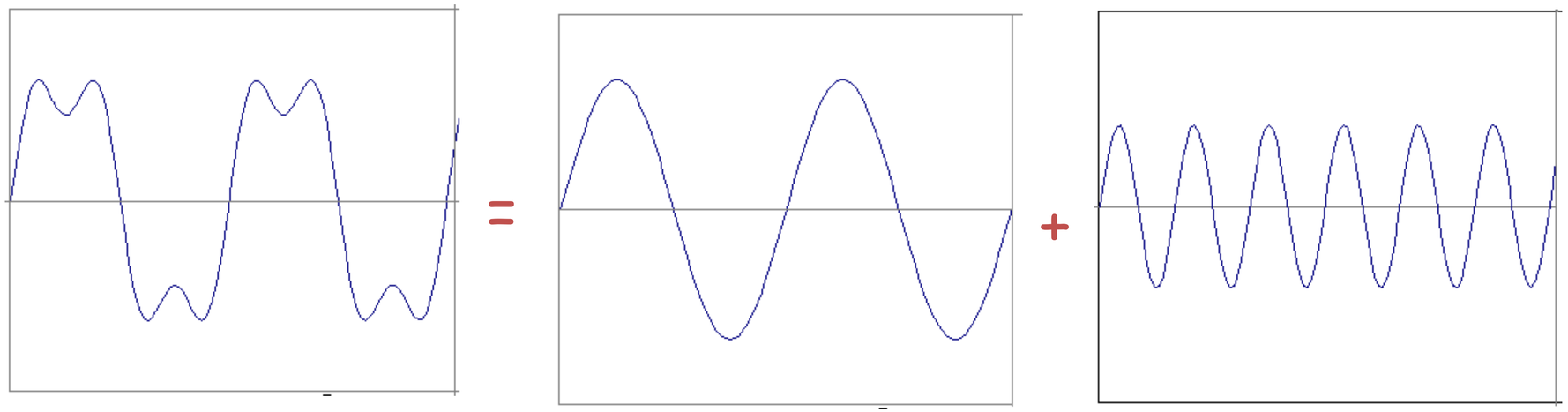
Discrete Fourier Transform (DFT) is a linear operator. Therefore, we can write:

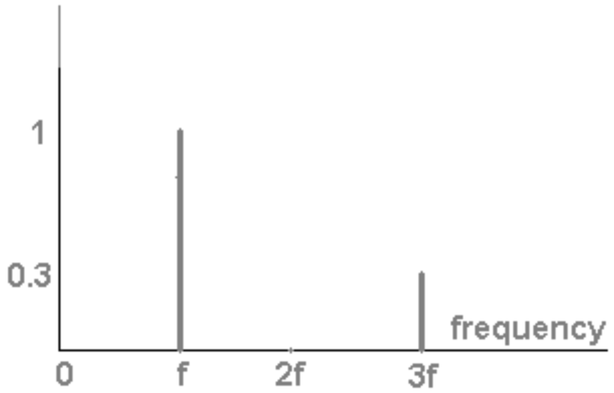


Source: Torralba, Freeman, Isola

Change of basis

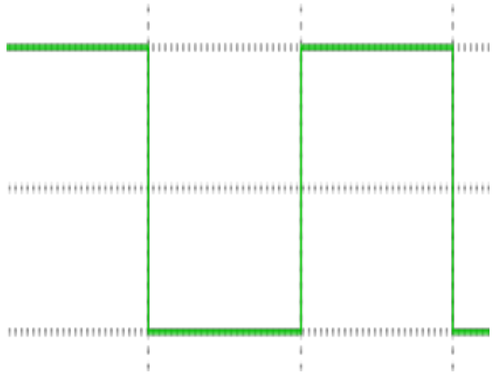
Example: $g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi(3f)\, t)$





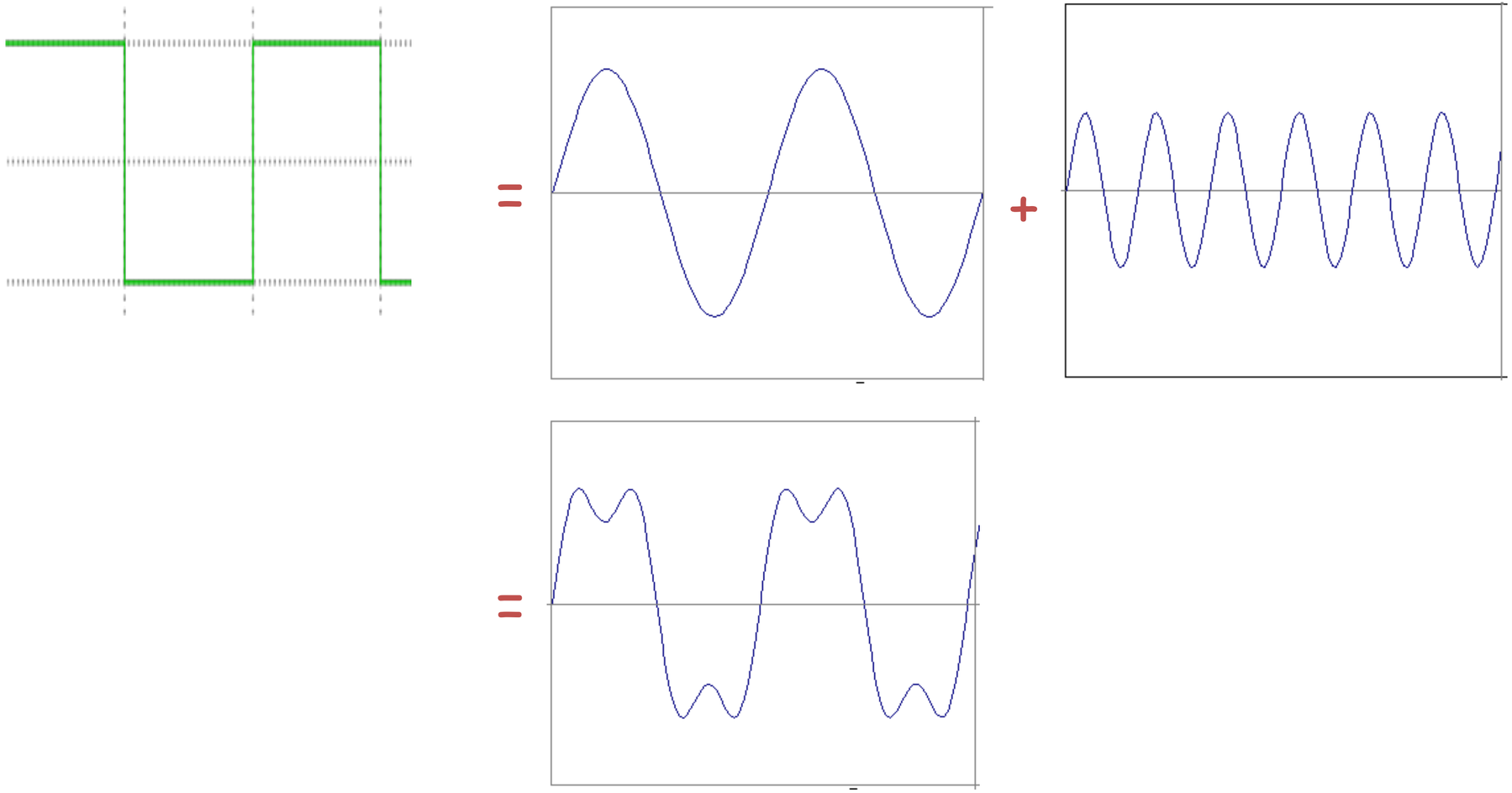
Source: A. Efros

Change of basis



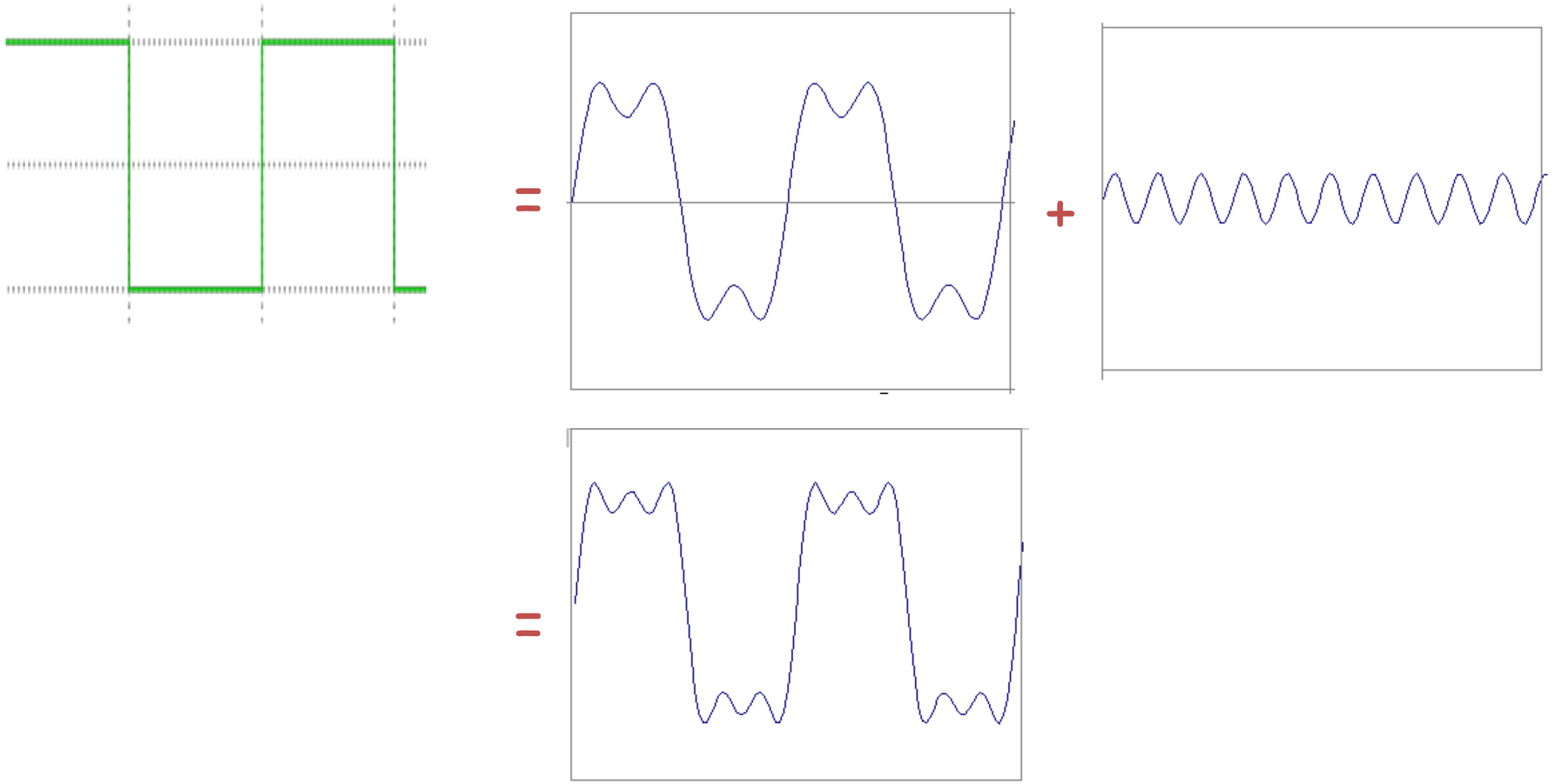
Source: A. Efros

Change of basis



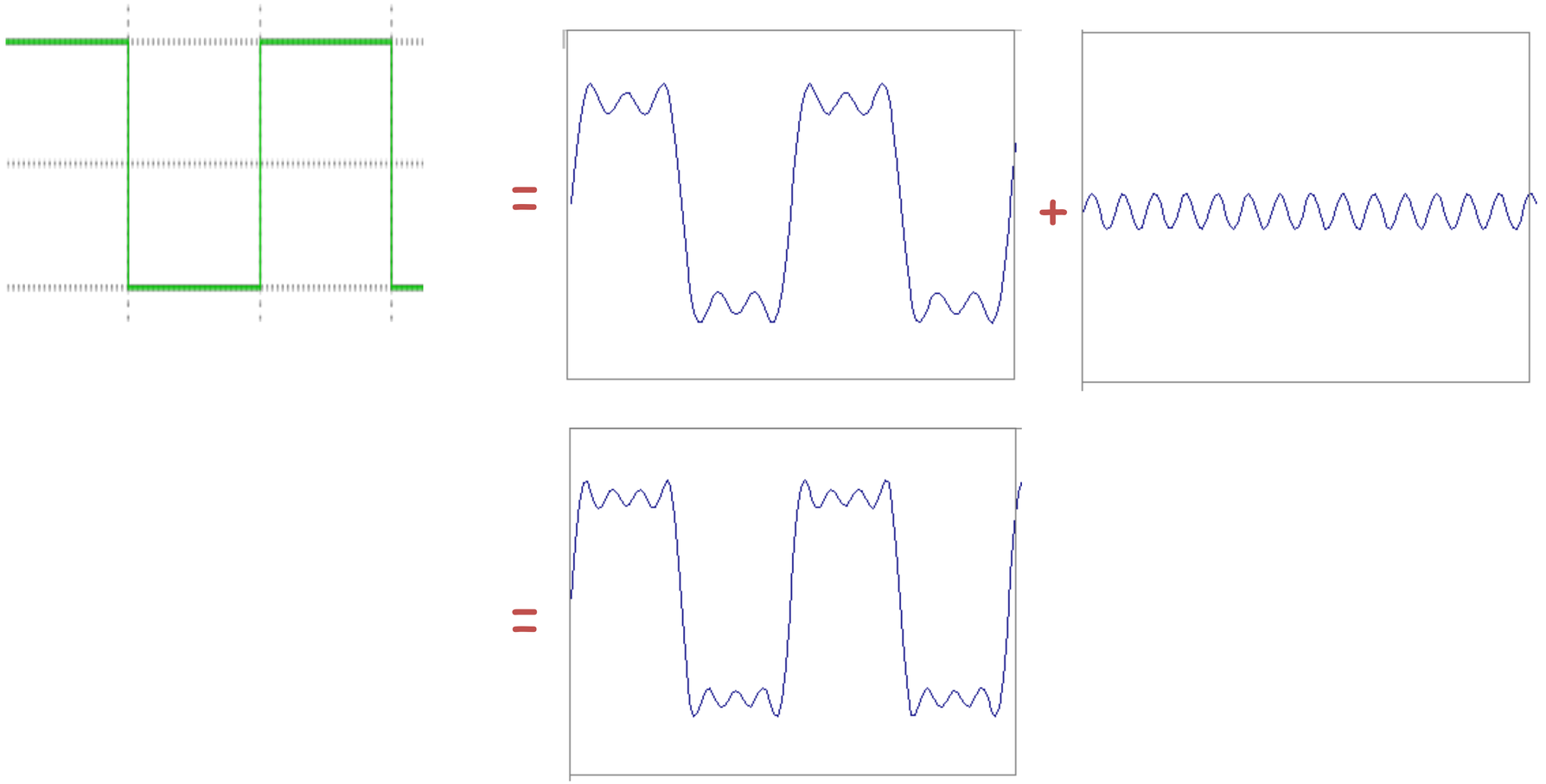
Source: A. Efros

Change of basis



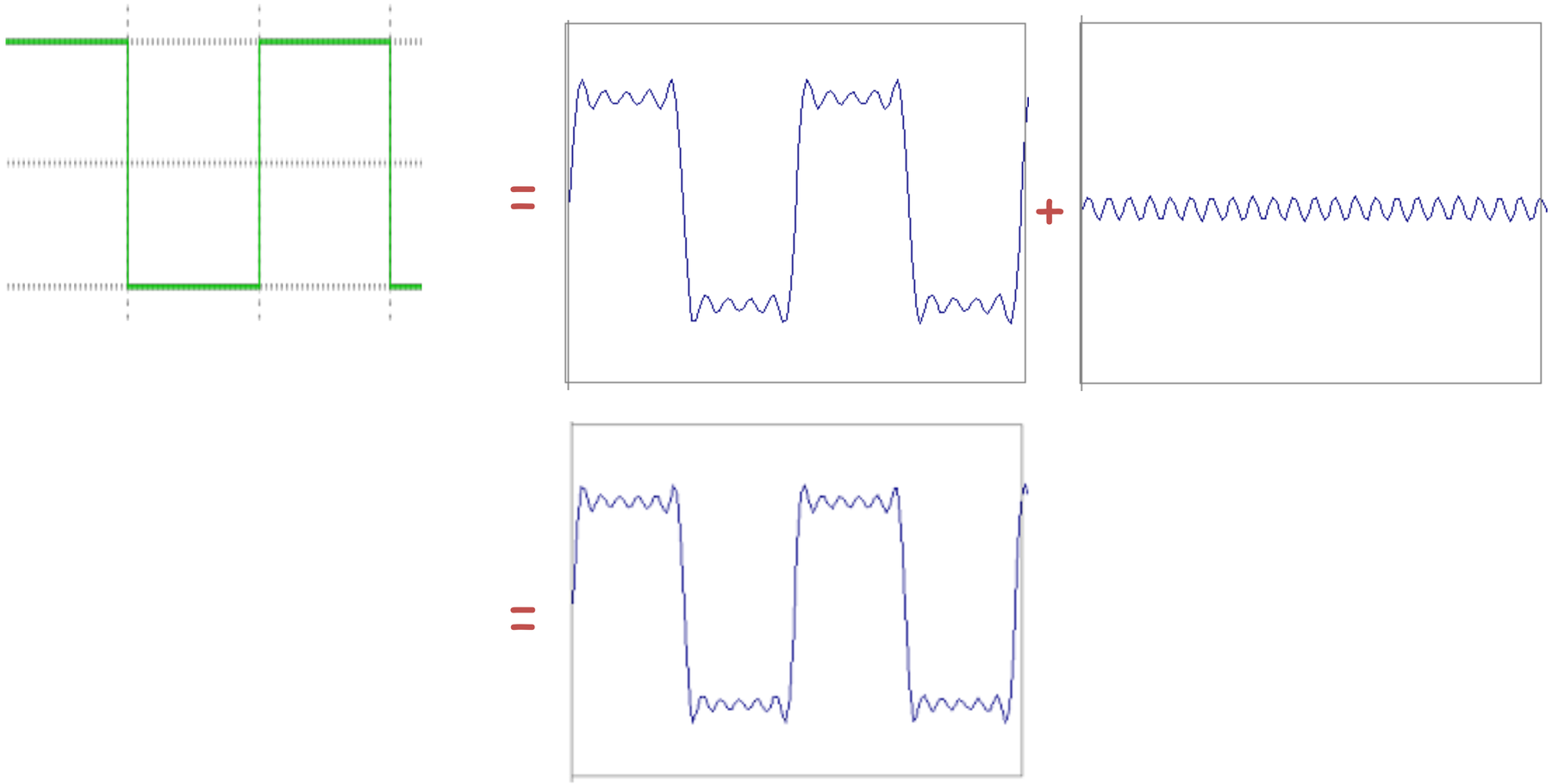
Source: A. Efros

Change of basis



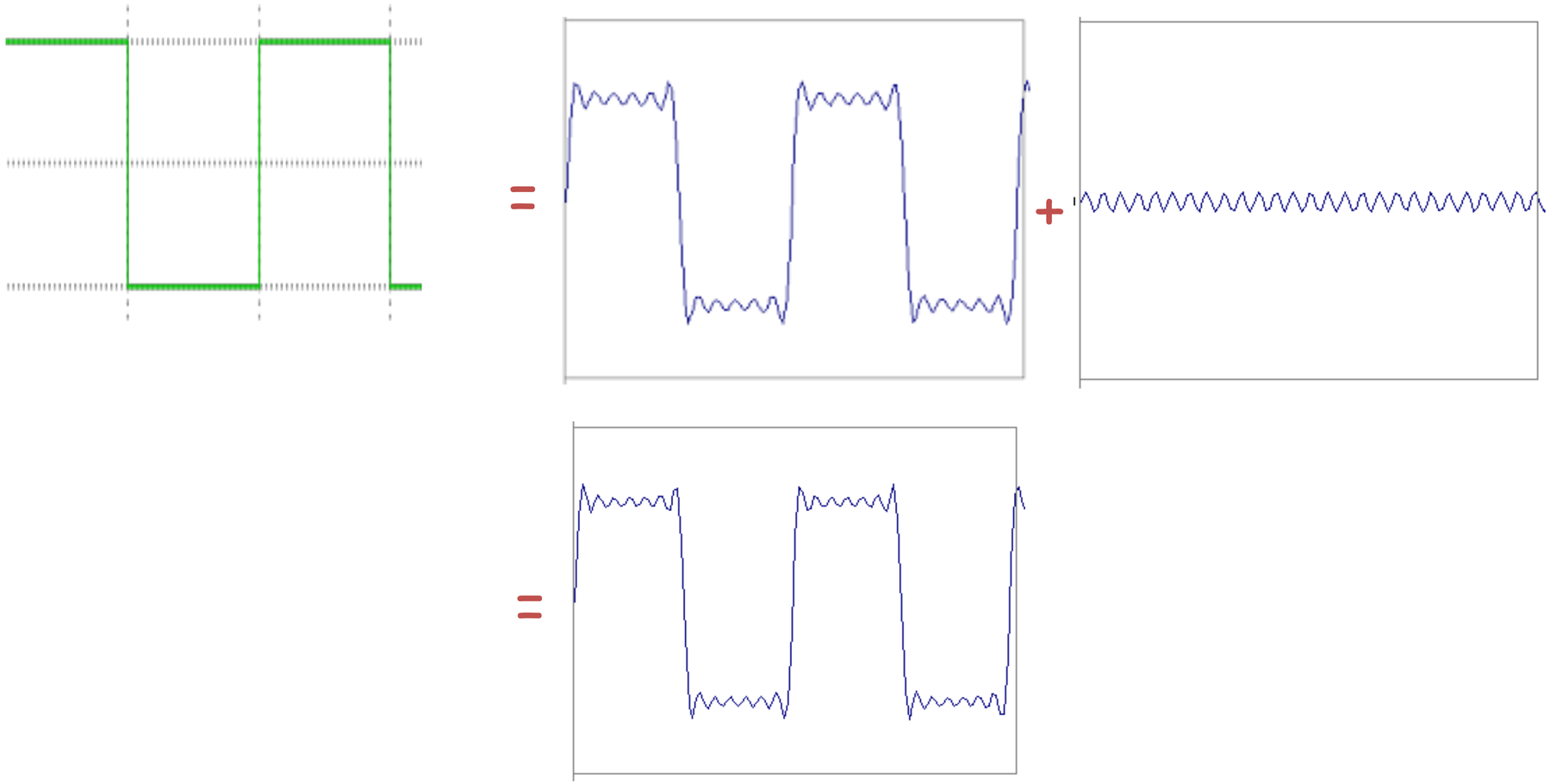
Source: A. Efros

Change of basis



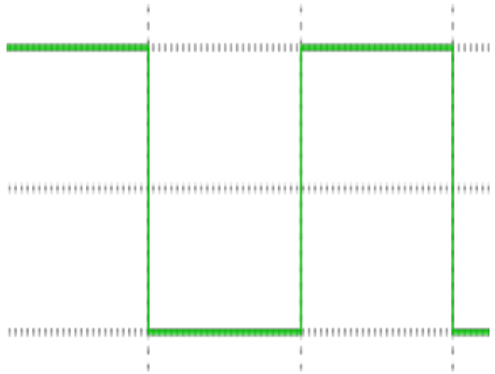
Source: A. Efros

Change of basis

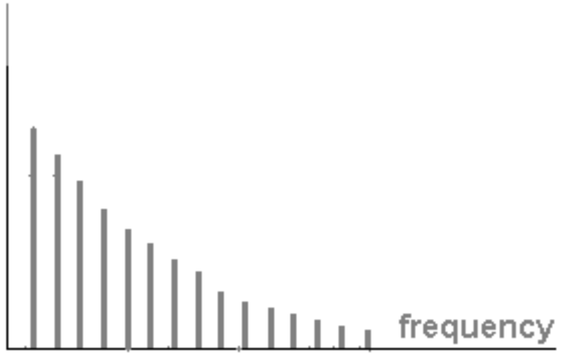


Source: A. Efros

Frequency Spectra



$$= \quad A \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi k t)$$



Source: A. Efros

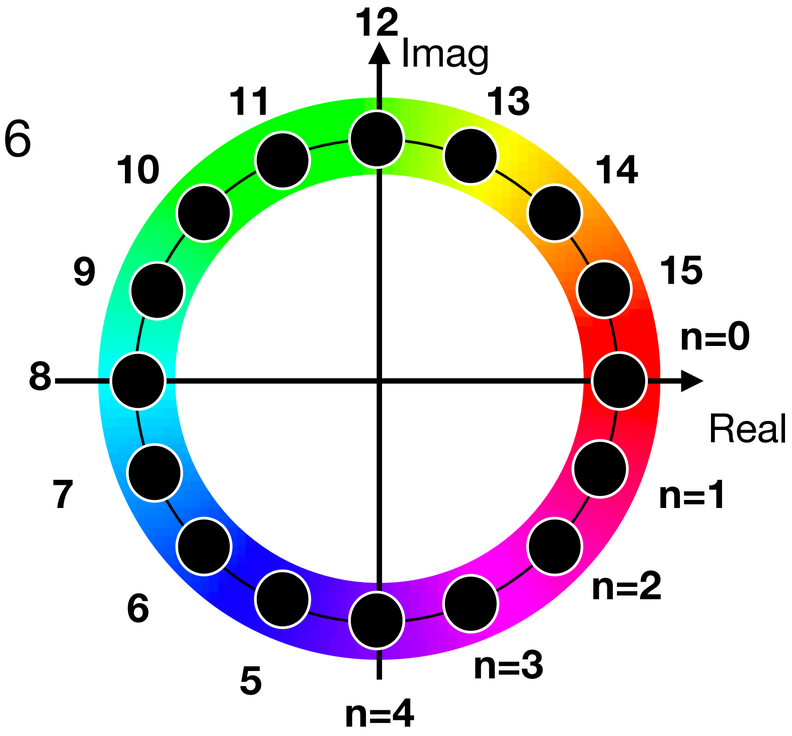
Visualizing the Fourier transform

$$F[u] = \sum_{n=0}^{N-1} f[n] \exp\left(-2\pi j \frac{un}{N}\right)$$

$$\exp(\alpha\, j) = \cos(\alpha) + j\, \sin(\alpha)$$

$$\cos\left(2\pi \frac{un}{N}\right) - j\ \sin\left(2\pi \frac{un}{N}\right)$$

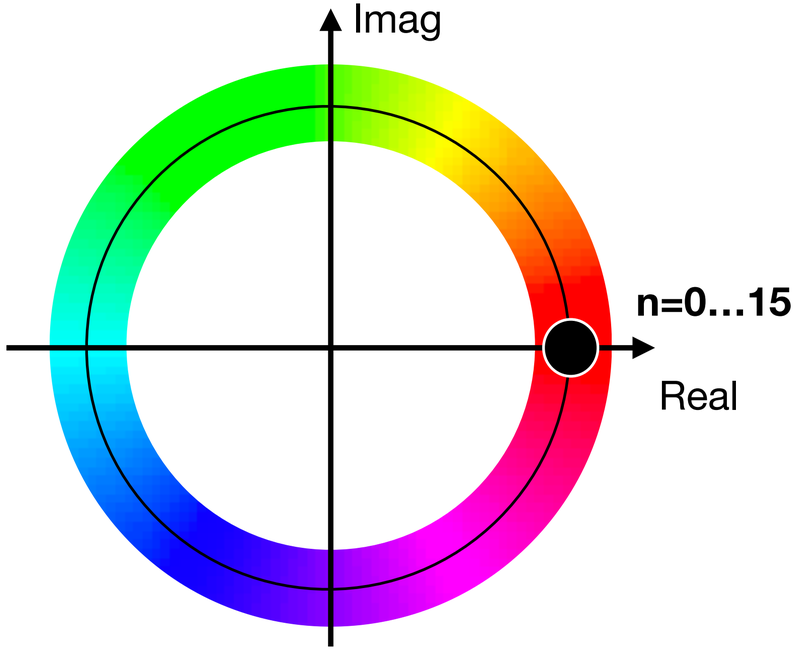
For:
u=1
N=16

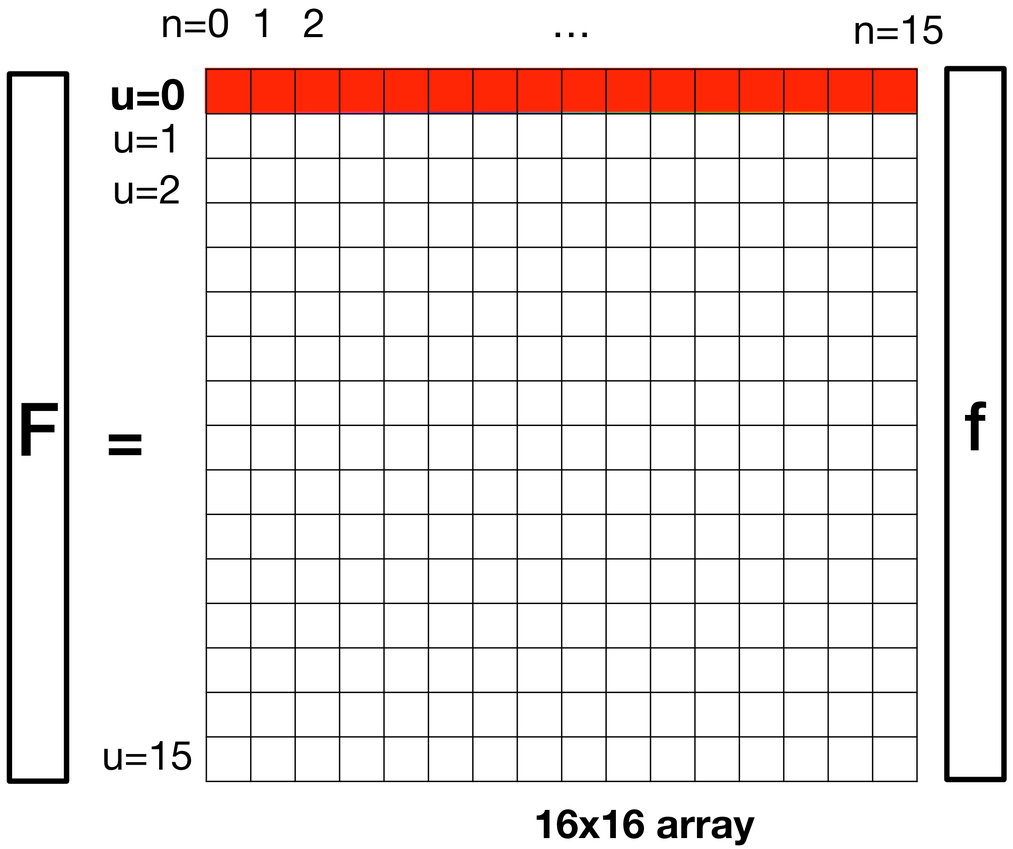


Source: Torralba, Freeman, Isola

Visualizing the transform coefficients

$$\exp\left(-2\pi j\frac{un}{N}\right)$$ For N=16

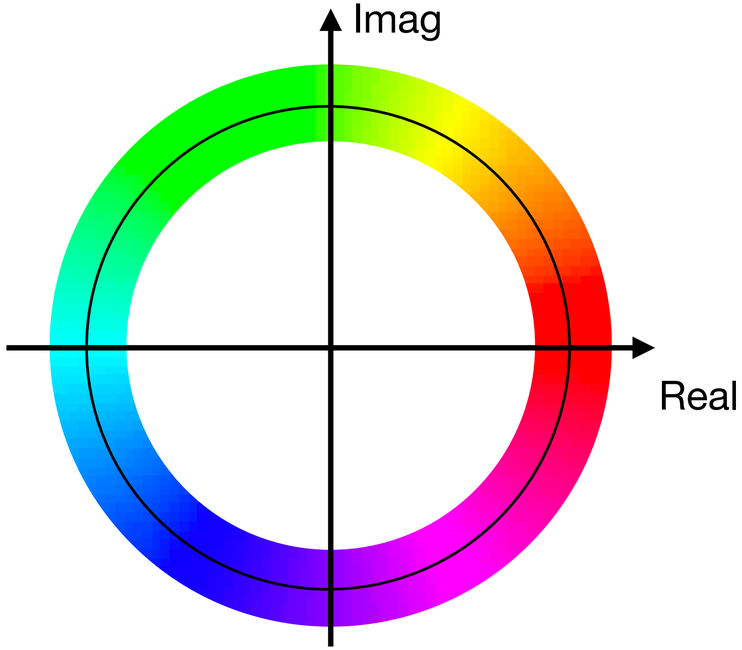


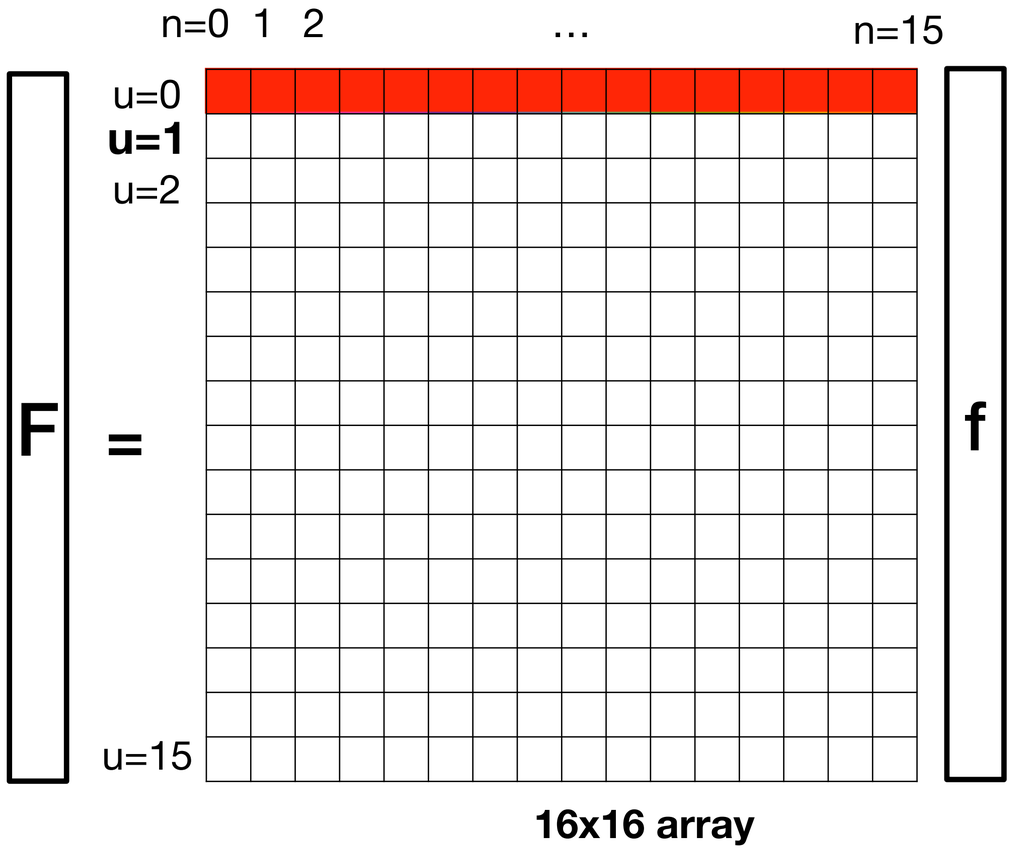


Source: Torralba, Freeman, Isola

Visualizing the transform coefficients

$$\exp\left(-2\pi j\frac{un}{N}\right)$$ For N=16

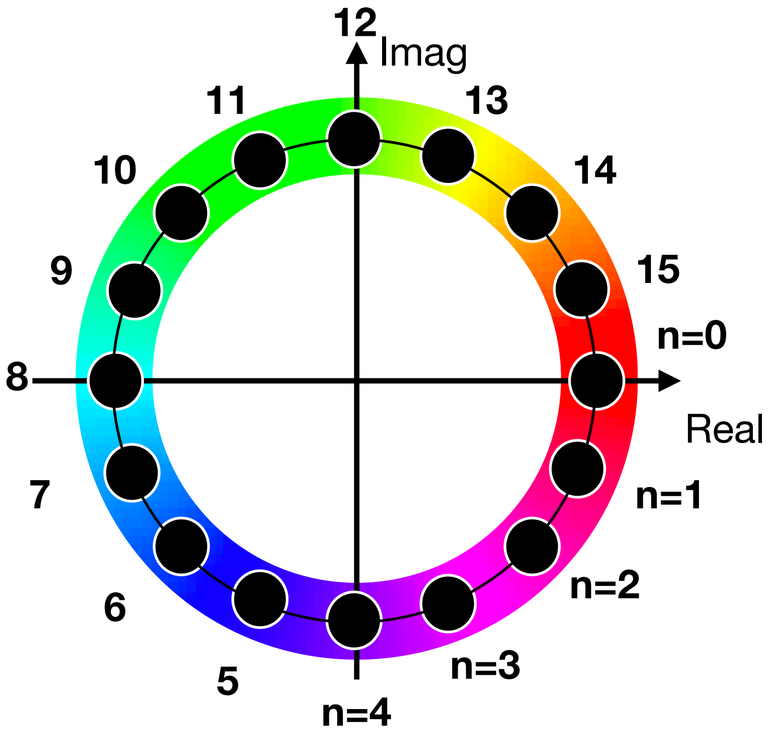


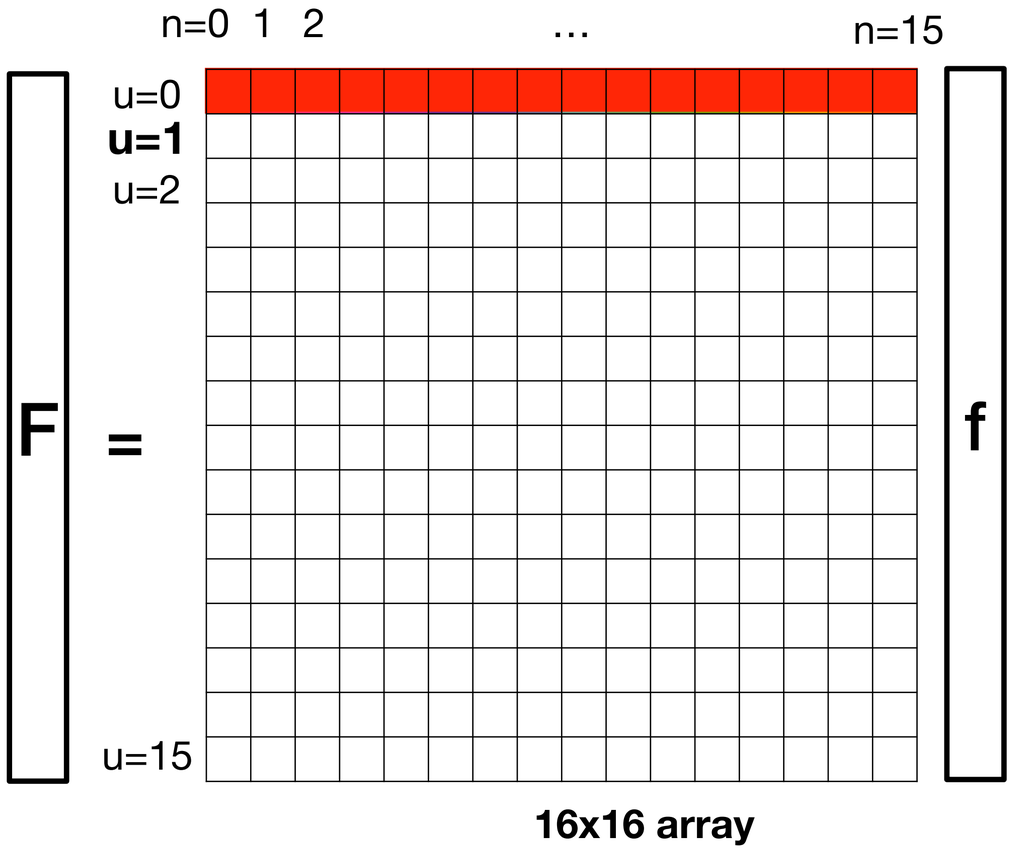


Source: Torralba, Freeman, Isola

Visualizing the transform coefficients

$$\exp\left(-2\pi j\frac{un}{N}\right)$$ For N=16

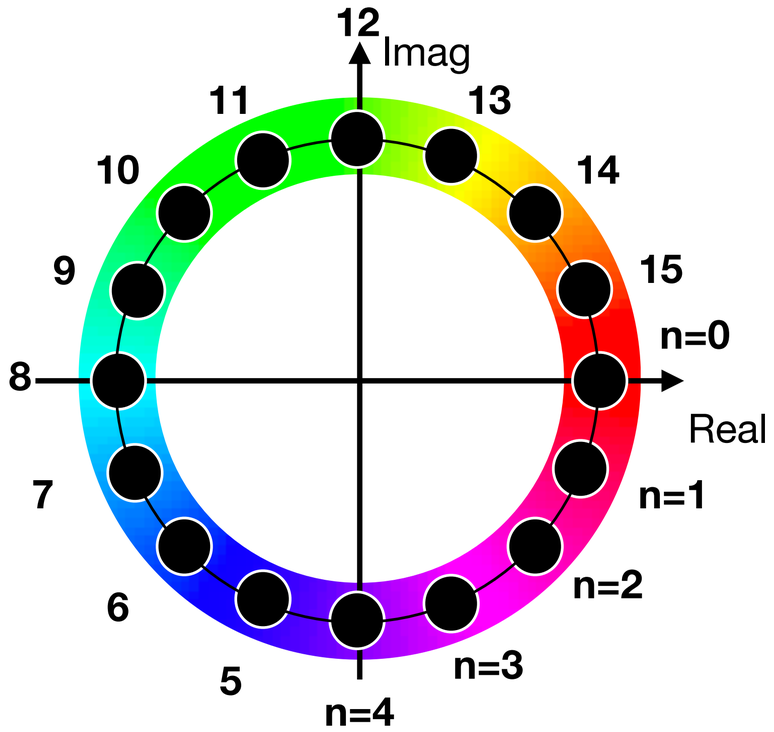


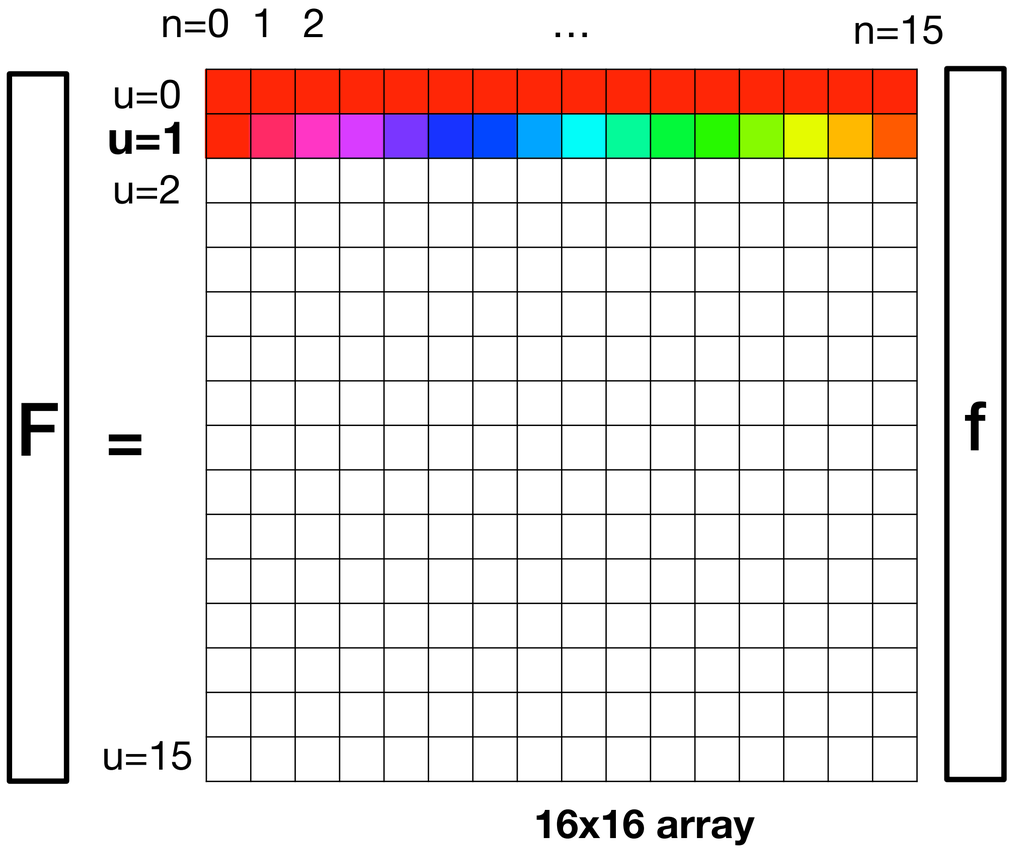


Source: Torralba, Freeman, Isola

Visualizing the transform coefficients

$$\exp\left(-2\pi j\frac{un}{N}\right)$$ For N=16

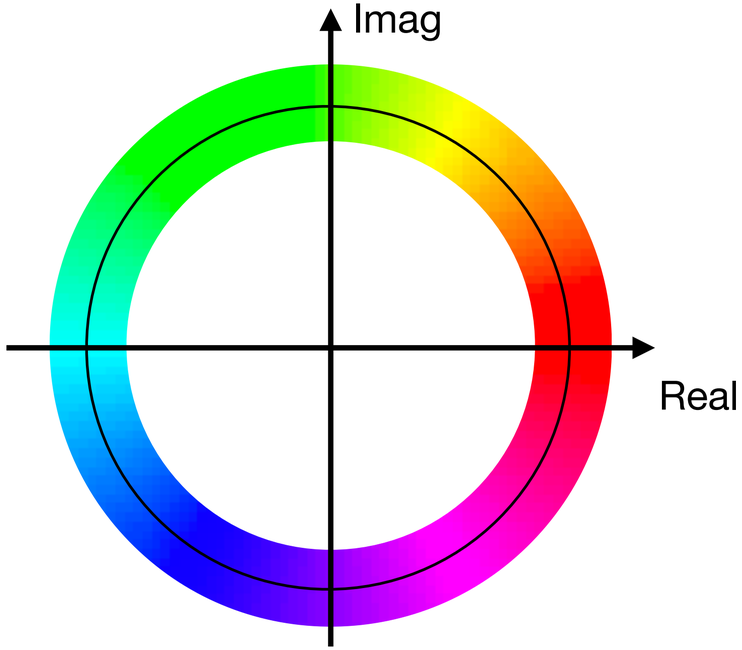


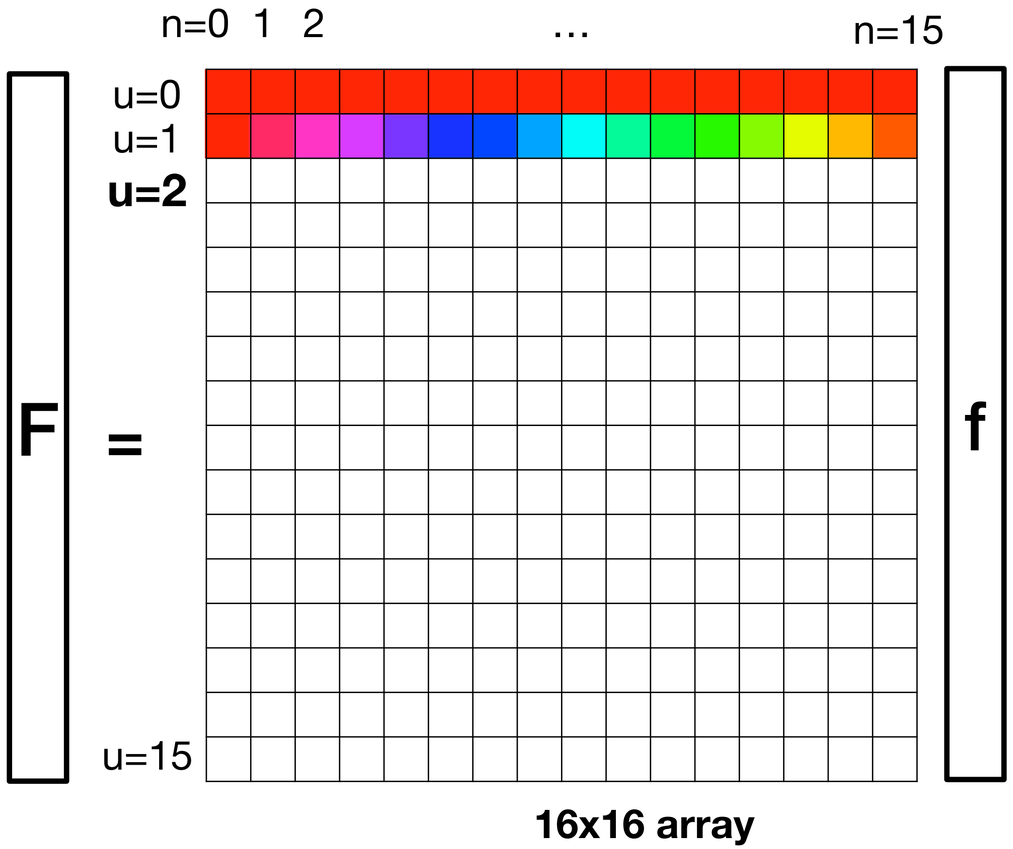


Source: Torralba, Freeman, Isola

Visualizing the transform coefficients

$$\exp\left(-2\pi j\frac{un}{N}\right)$$ For N=16

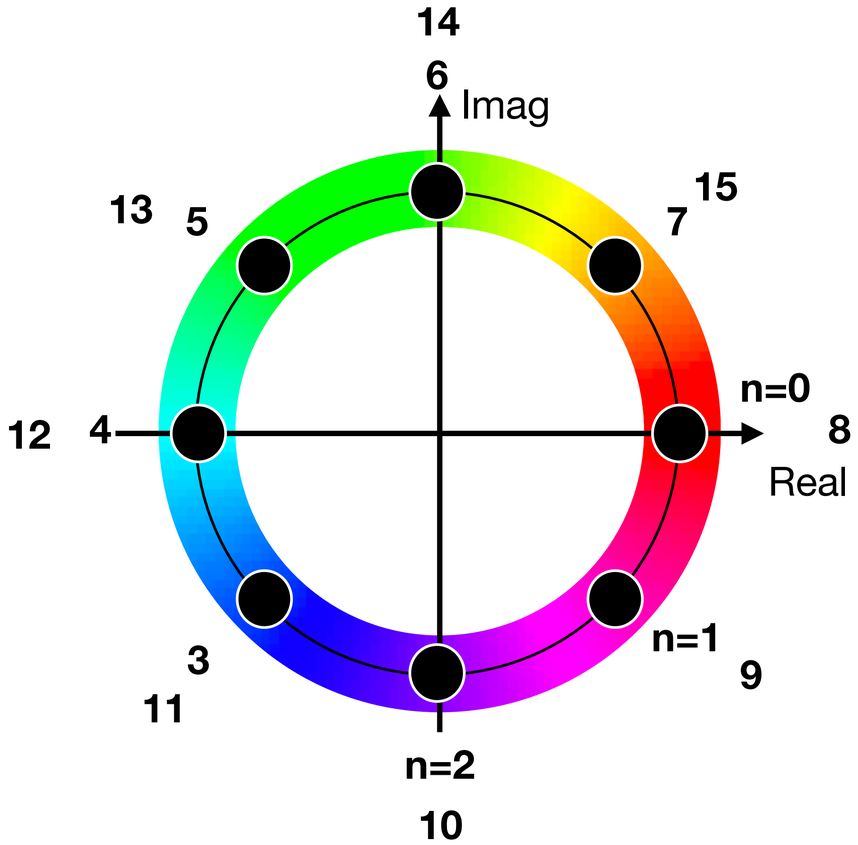


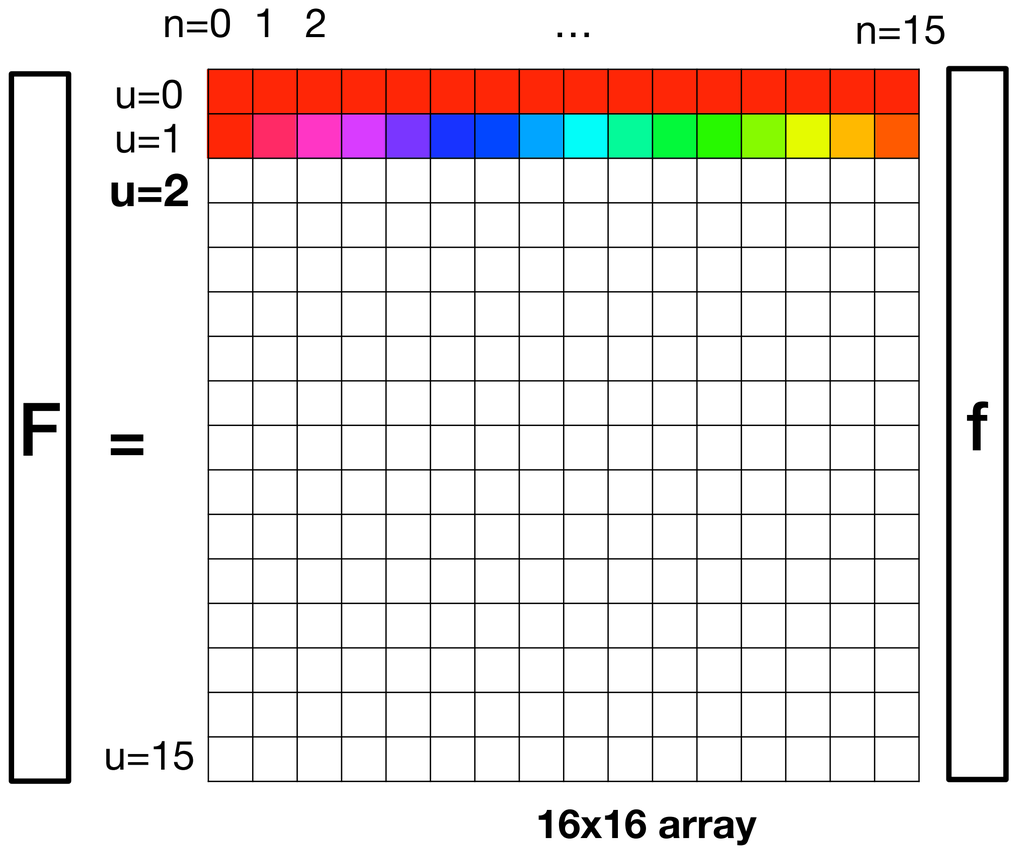


Source: Torralba, Freeman, Isola

Visualizing the transform coefficients

$$\exp\left(-2\pi j\frac{un}{N}\right)$$ For N=16

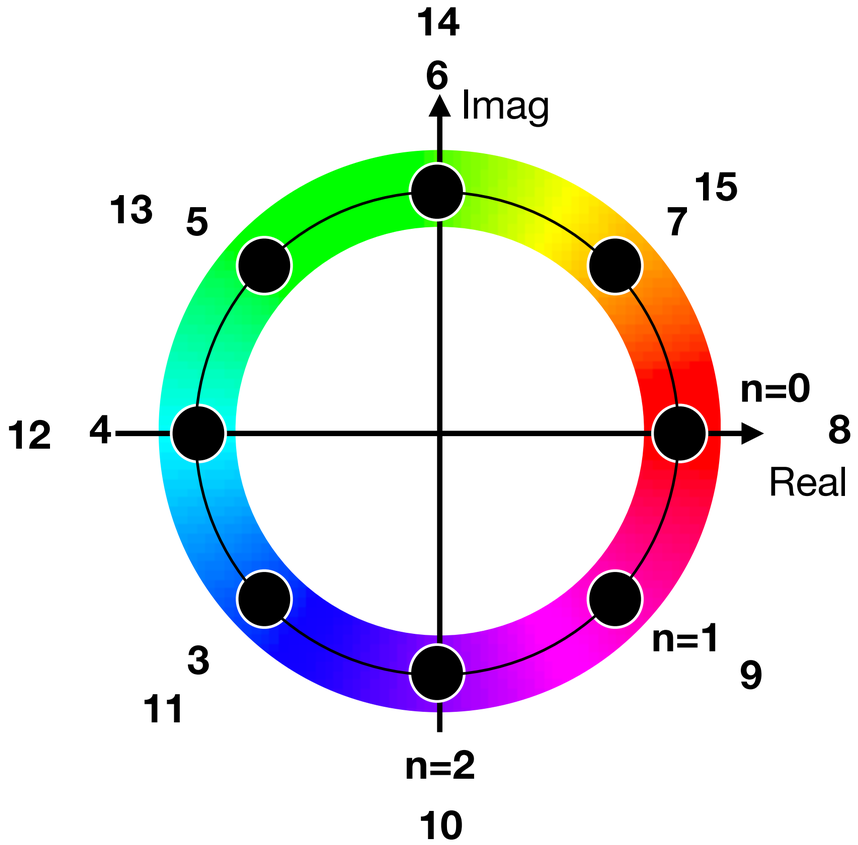


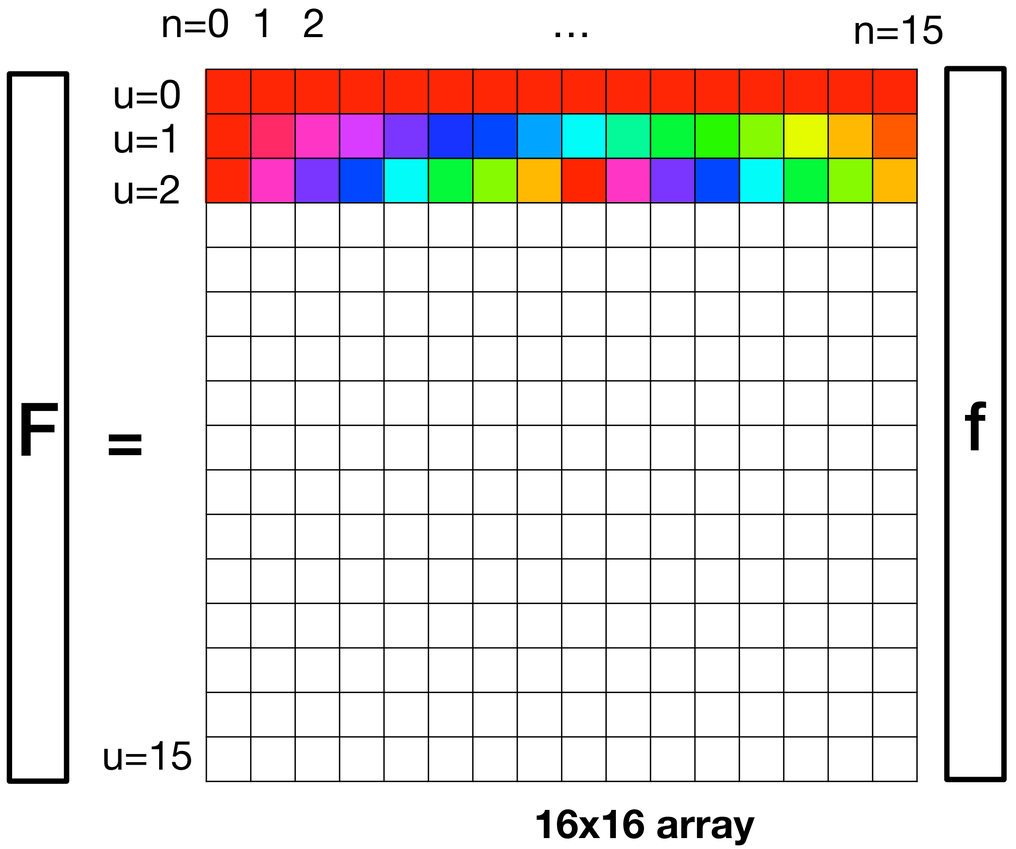


Source: Torralba, Freeman, Isola

Visualizing the transform coefficients

$$\exp\left(-2\pi j\frac{un}{N}\right)$$ For N=16



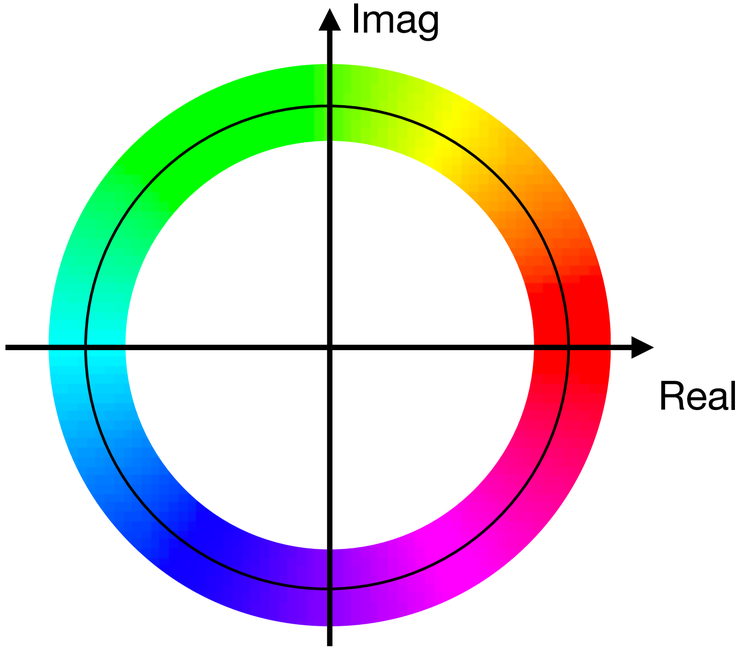


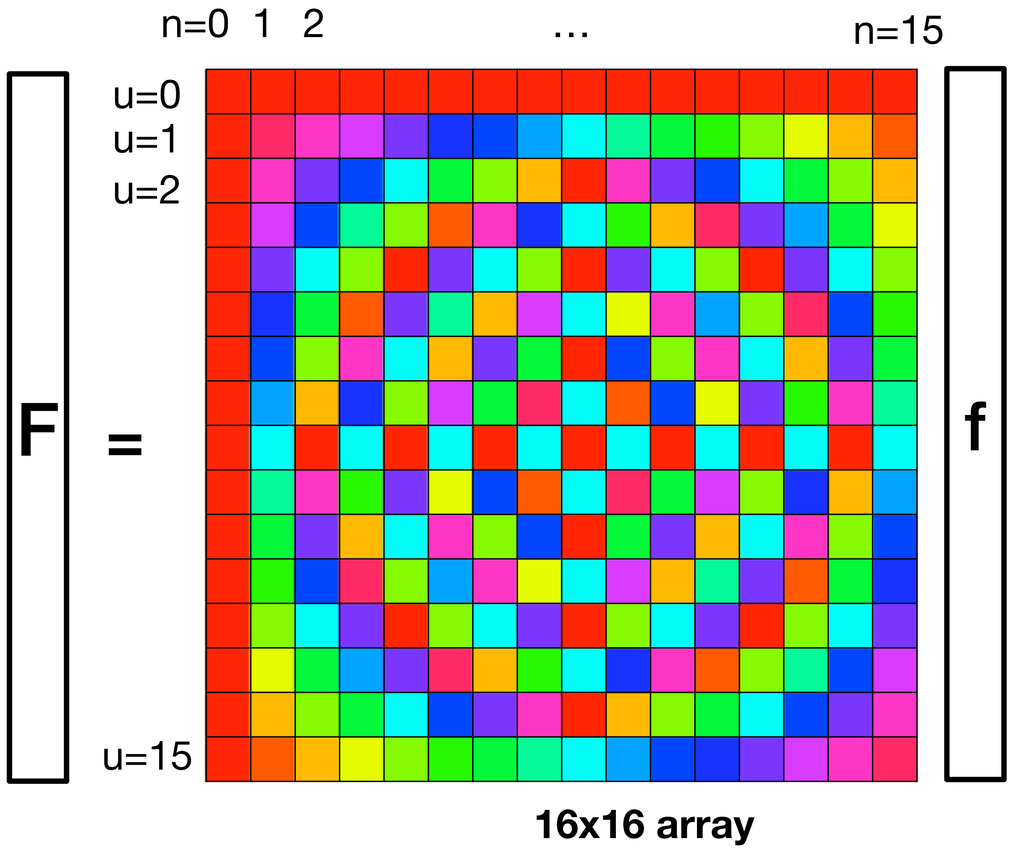
Source: Torralba, Freeman, Isola

Visualizing the transform coefficients

$$\exp\left(-2\pi j\frac{un}{N}\right)$$

For N=16





Source: Torralba, Freeman, Isola

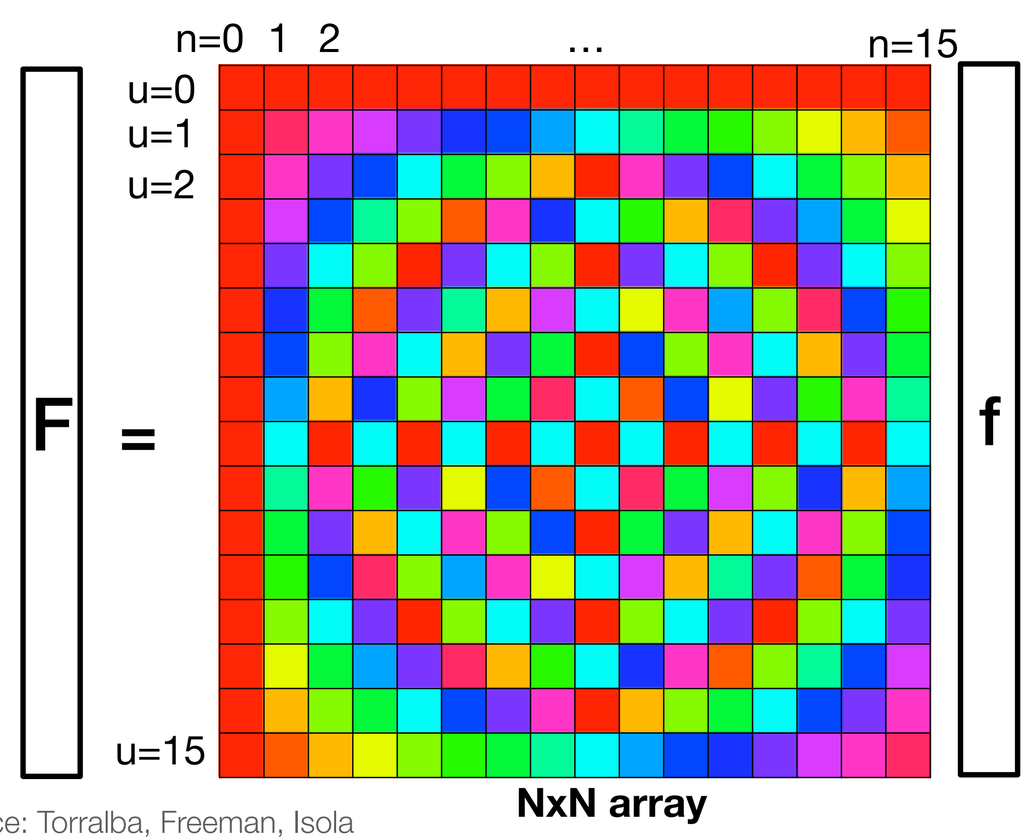
The inverse of the Discrete Fourier transform

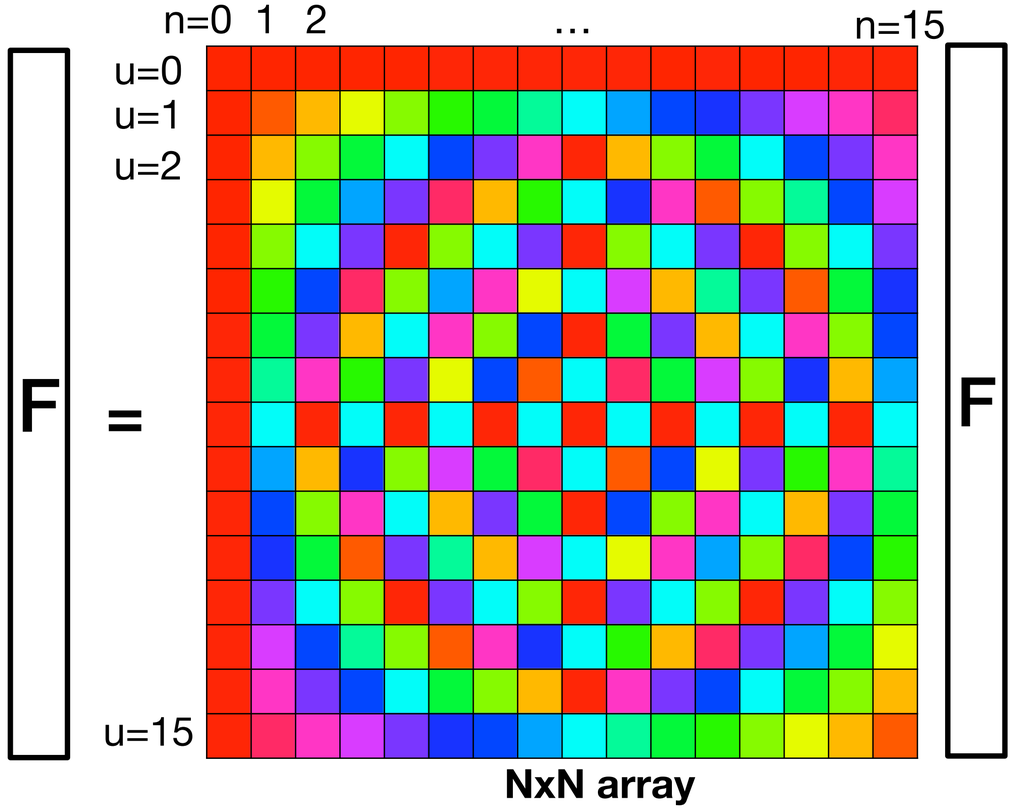
Discrete Fourier Transform (DFT):

$$F[u] = \sum_{n=0}^{N-1} f[n] \exp\left(-2\pi j \frac{un}{N}\right)$$

Its inverse:

$$f[n] = \frac{1}{N} \sum_{u=0}^{N-1} F[u] \exp\left(2\pi j \frac{un}{N}\right)$$





Source: Torralba, Freeman, Isola

For images, the 2D DFT

1D Discrete Fourier Transform (DFT) transforms a signal $f[n]$ into $F[u]$ as:

$$F[u] = \sum_{n=0}^{N-1} f[n] \exp\left(-2\pi j \frac{un}{N}\right)$$

2D Discrete Fourier Transform (DFT) transforms an image $f[n,m]$ into $F[u,v]$ as:

$$F[u,v] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f[n,m] \exp\left(-2\pi j \left(\frac{un}{N} + \frac{vm}{M}\right)\right)$$

Source: Torralba, Freeman, Isola

Useful properties of the Fourier transform

- 2D Fourier transform is separable (just like Gaussian)
- Fast Fourier transform (FFT) computable in $O(n \log(n))$
- Complex exponentials are eigenfunctions of linear shift invariant systems
  - i.e. If we convolve a complex exponential *f* with filter *g*, we get a complex scaled version out:

$$f[u] = e^{j\omega u}, \qquad f * g = (a + bj)f$$

Useful properties of the Fourier transform

- Convolution is multiplication in the Fourier domain!

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}$$

- Can be much faster for big filters
- Speed is independent of filter size!

Visualizing the image Fourier transform

$$F[u,v] = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} f[n,m] \exp\left(-2\pi j\left(\frac{un}{N}+\frac{vm}{M}\right)\right)$$

The values of $F[u,v]$ are complex.

Using the real and imaginary components:

$$F[u,v] = Re\{F[u,v]\} + j\,Imag\{F[u,v]\}$$

Or using a polar decomposition:

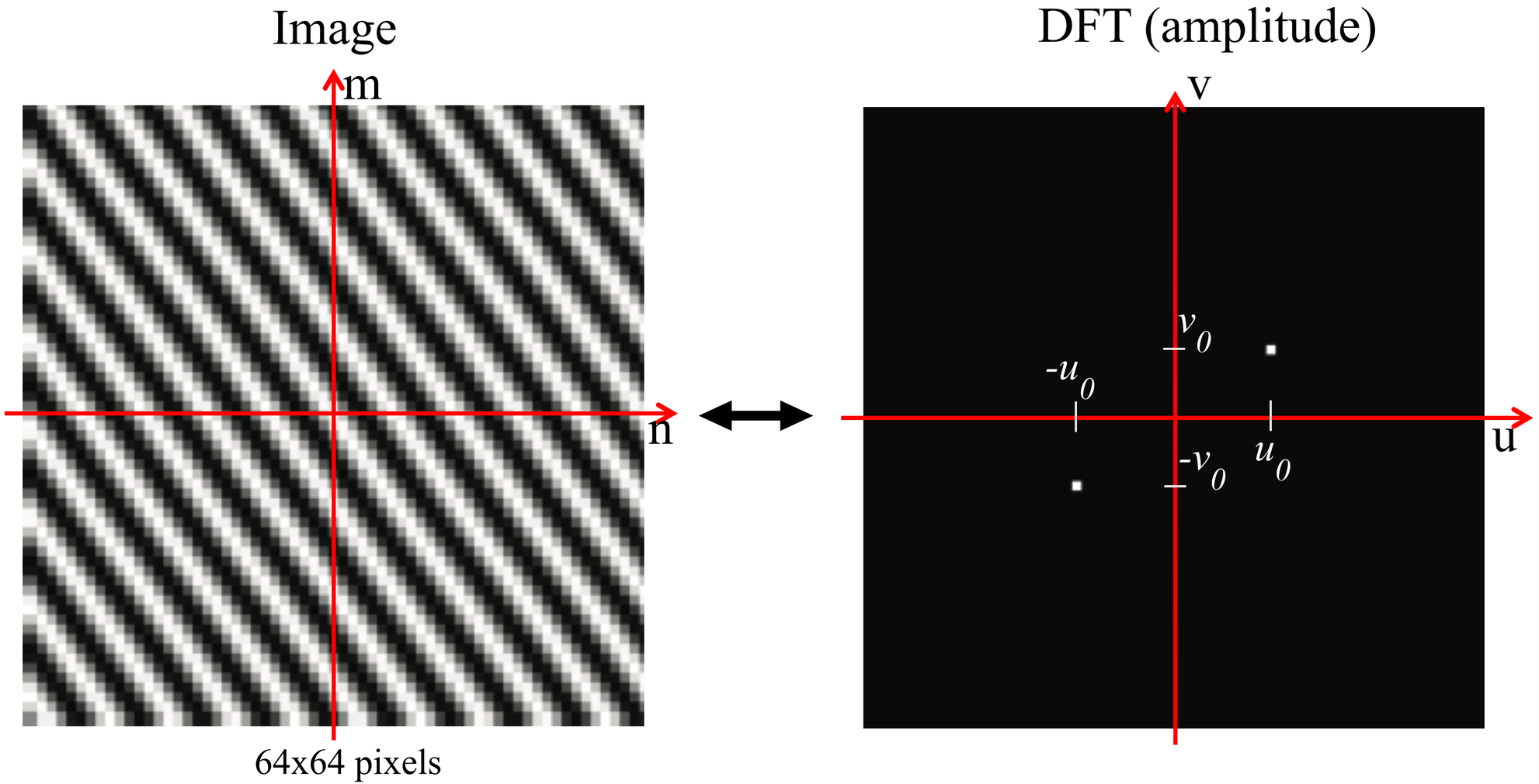
$$F[u,v] = A[u,v]\exp(j\theta[u,v])$$

Amplitude

Phase

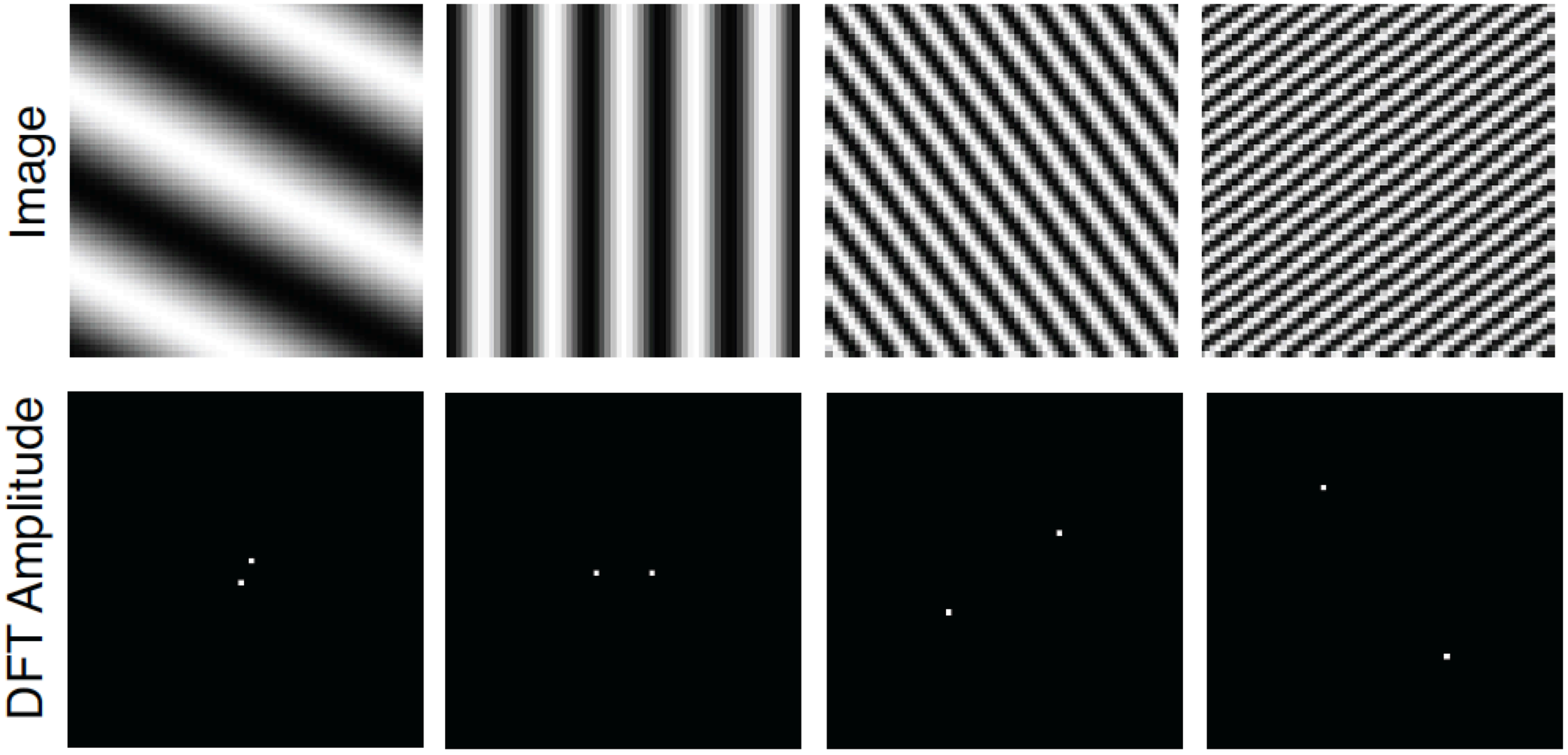
Source: Torralba, Freeman, Isola

Simple Fourier transforms



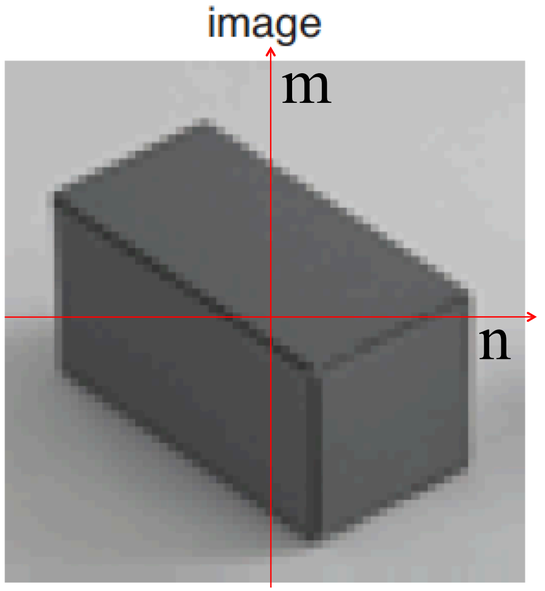
$$\cos\left(2\pi\left(\frac{u_0\,n}{N}+\frac{v_0\,m}{M}\right)\right) \quad \longleftrightarrow \quad \frac{1}{2}\left(\delta\left[u-u_0,v-v_0\right]+\delta\left[u+u_0,v+v_0\right]\right)$$

Simple Fourier transforms

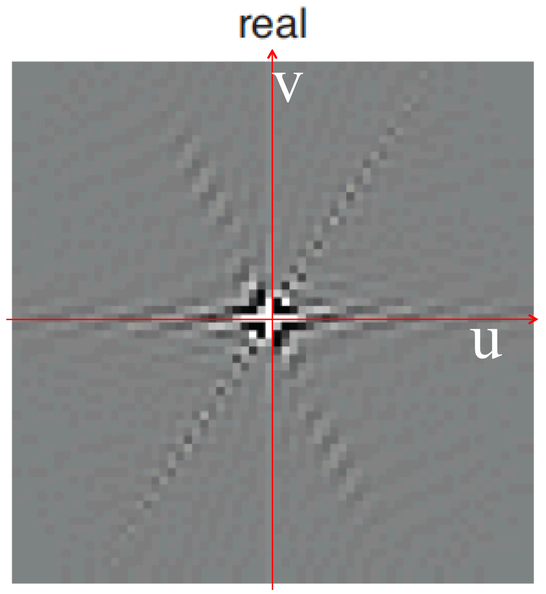


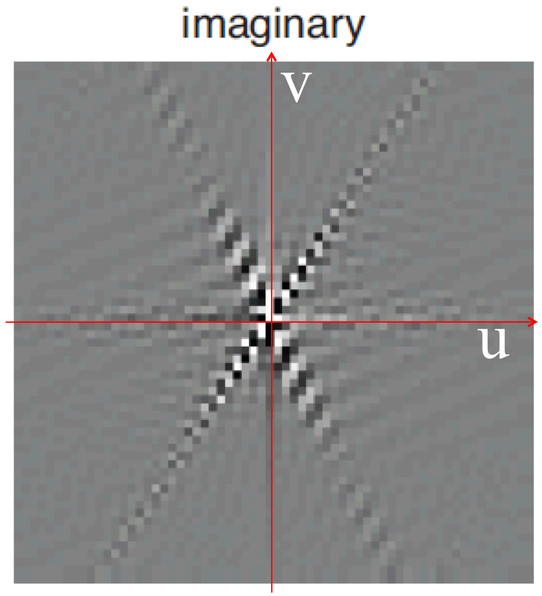
Visualizing the image Fourier transform

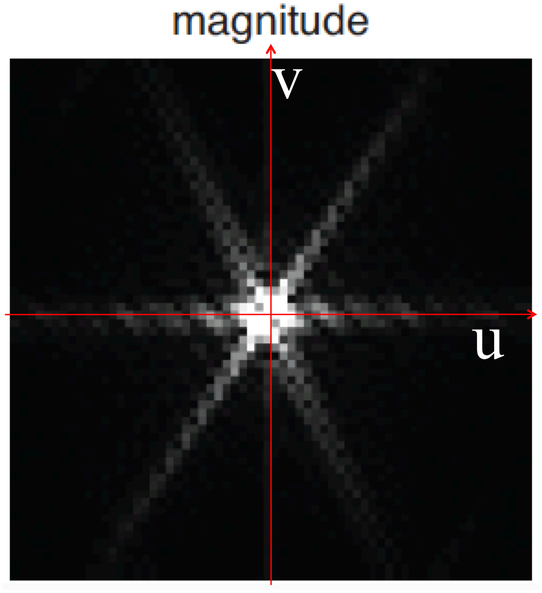
$f[n, m]$

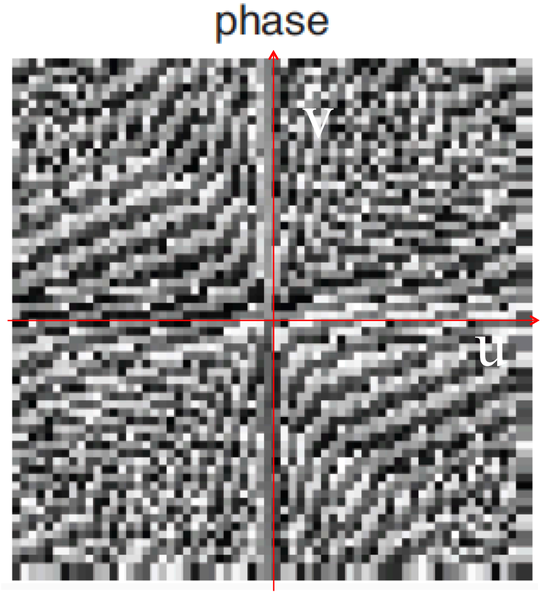


$F[u, v]$

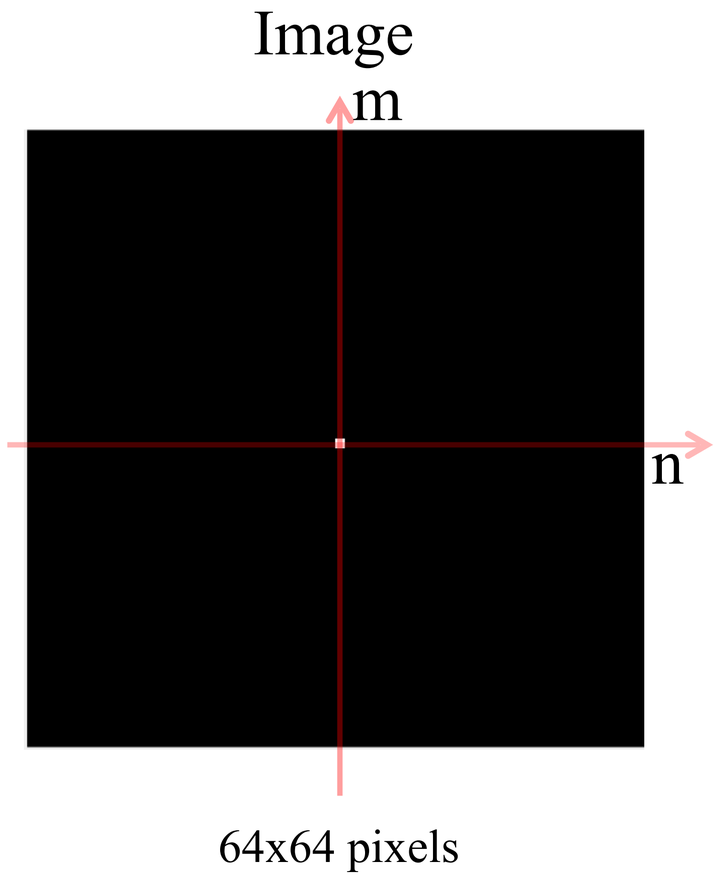


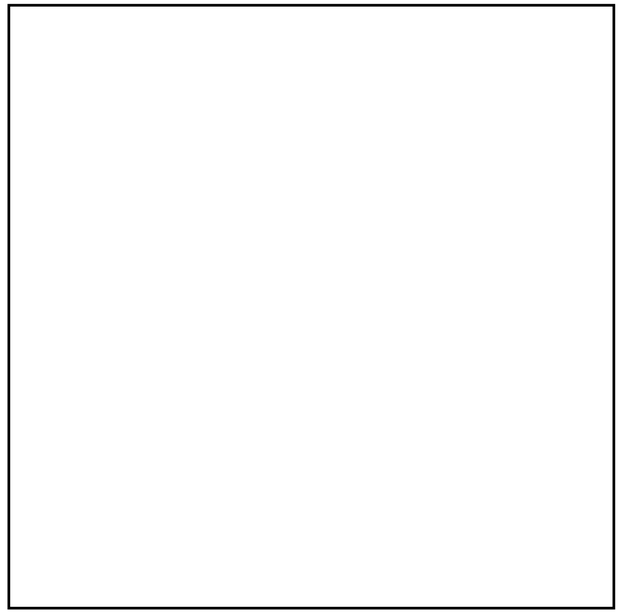




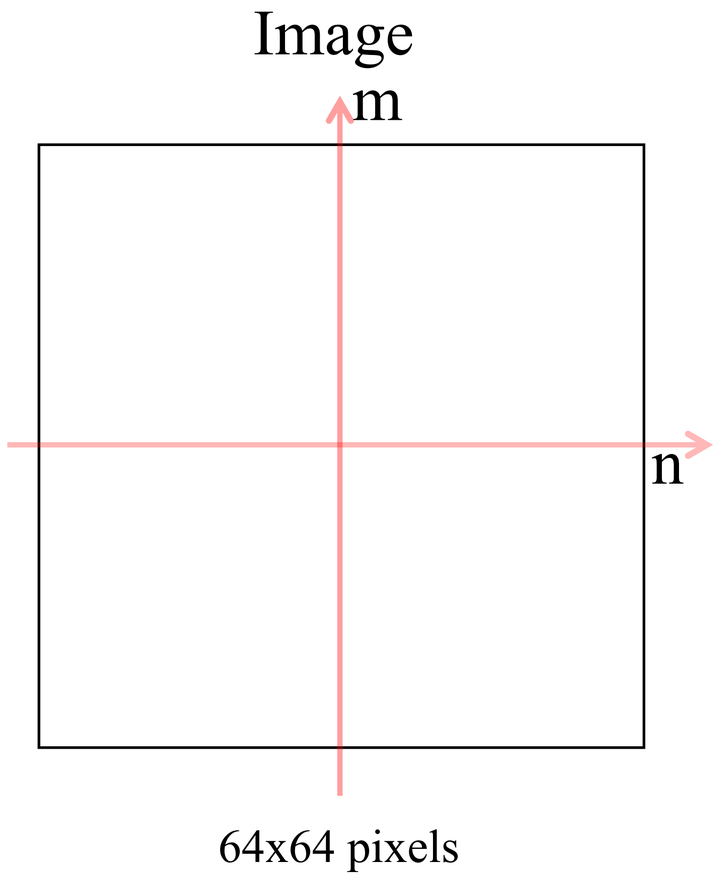


Some important Fourier transforms

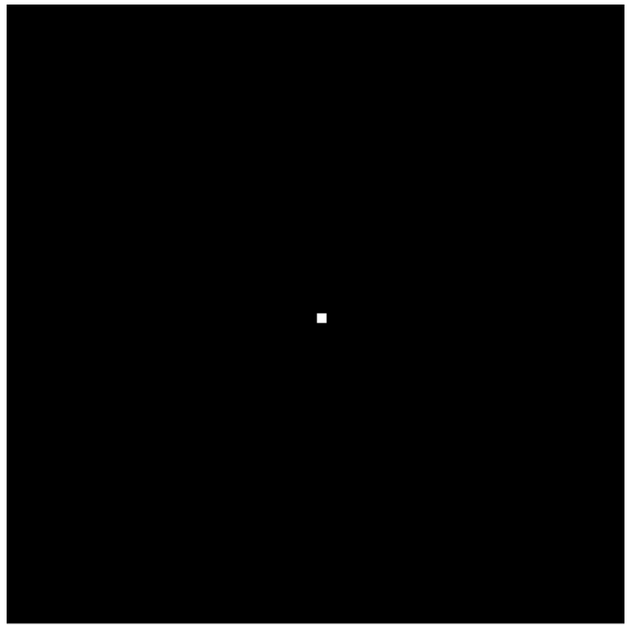




Some important Fourier transforms

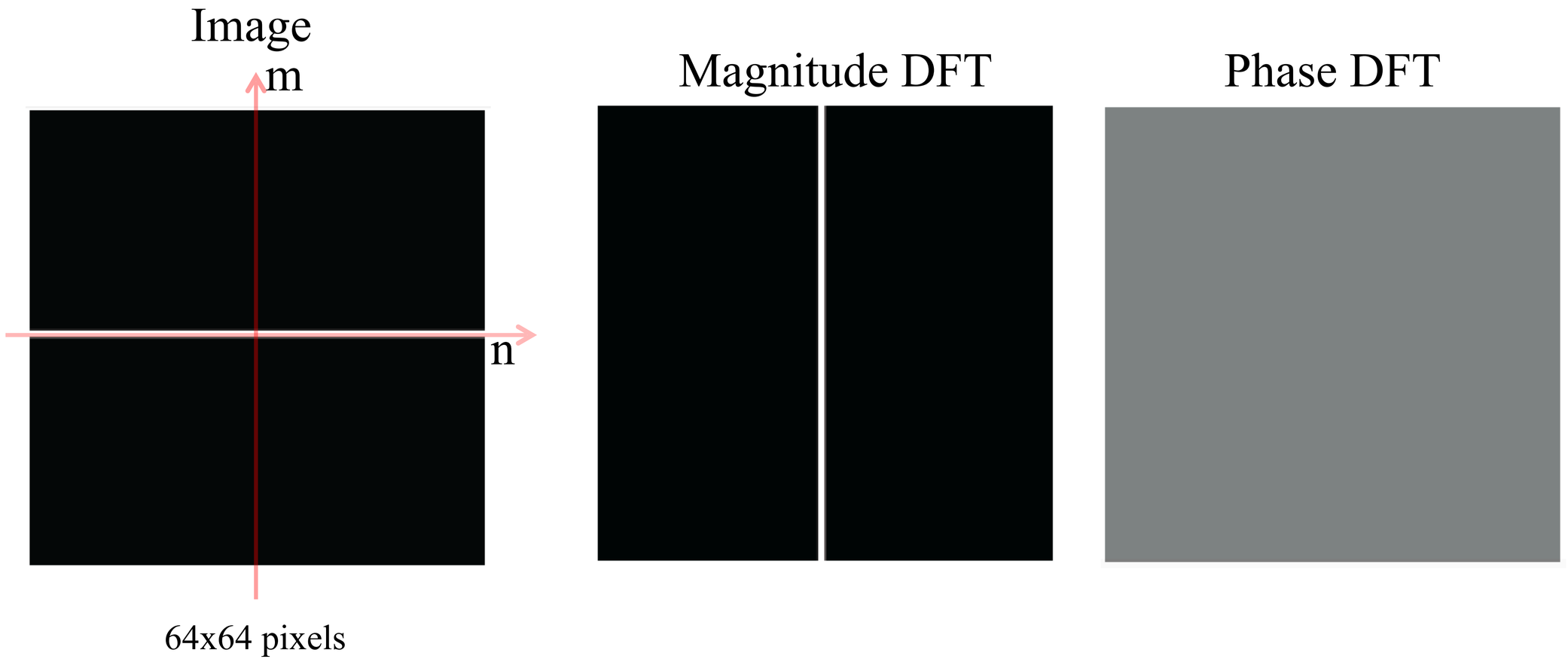


Magnitude DFT

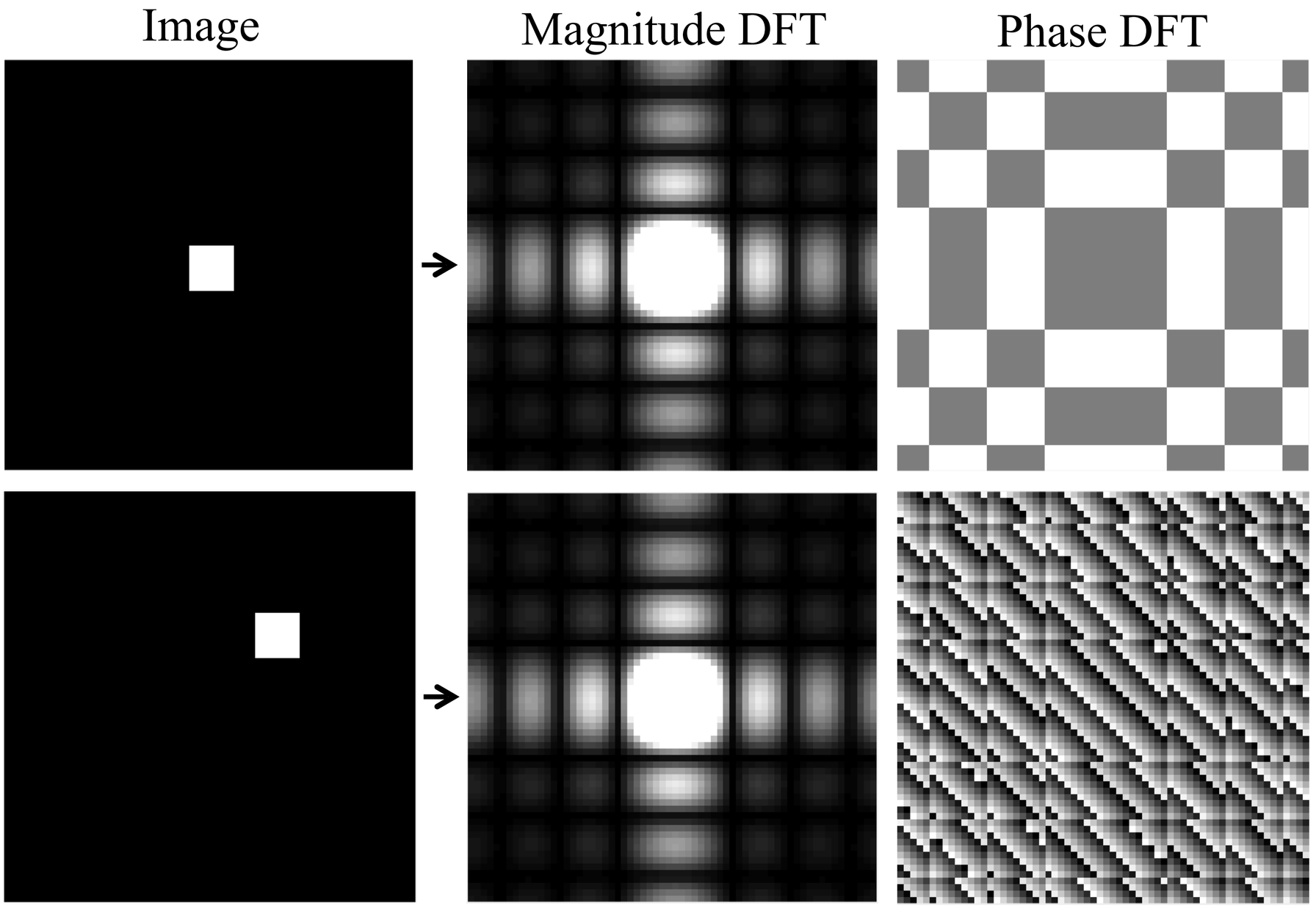


Phase DFT

Some important Fourier transforms



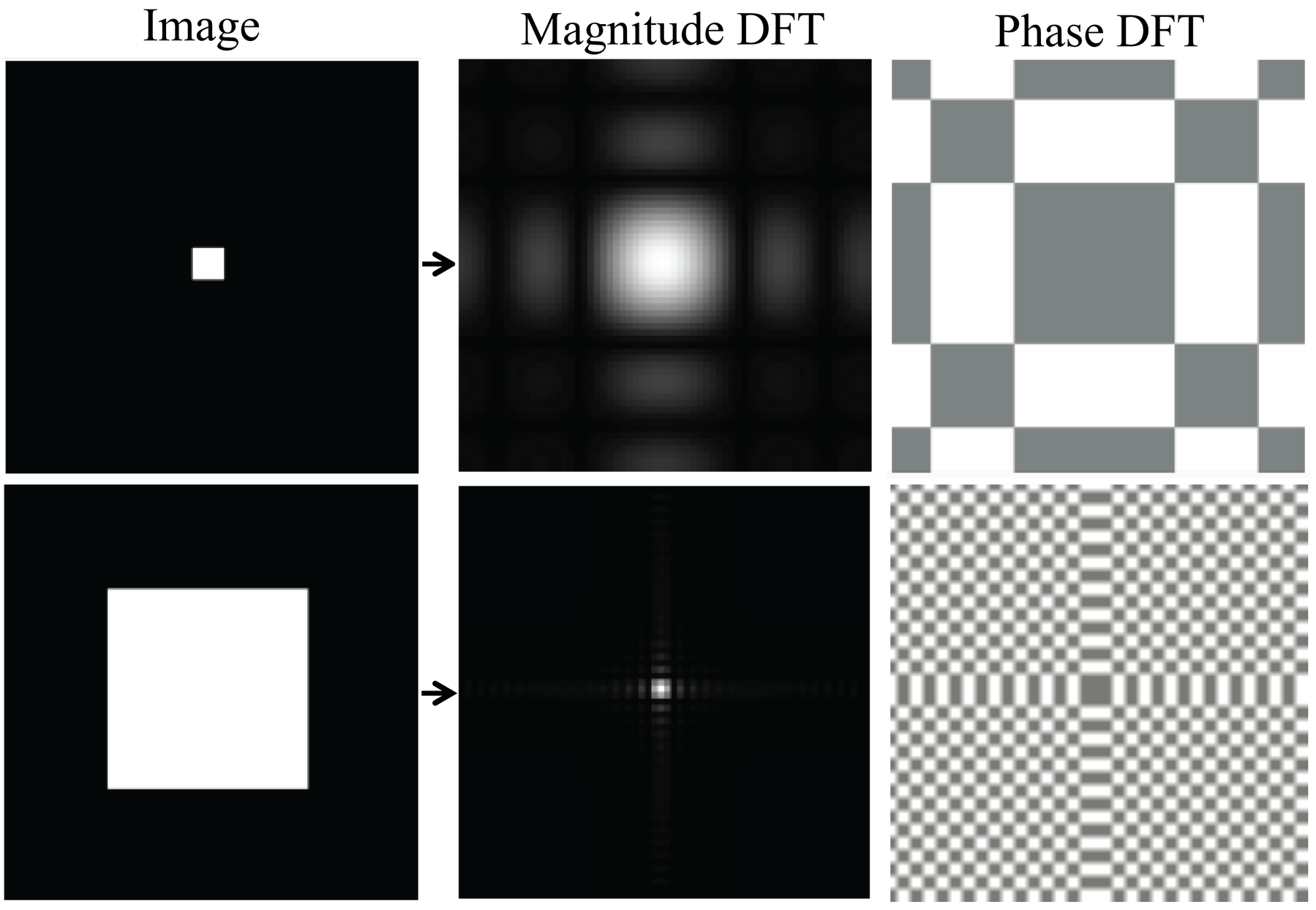
Some important Fourier transforms



Translation

Shifts of an image only produce changes on the phase of the DFT.

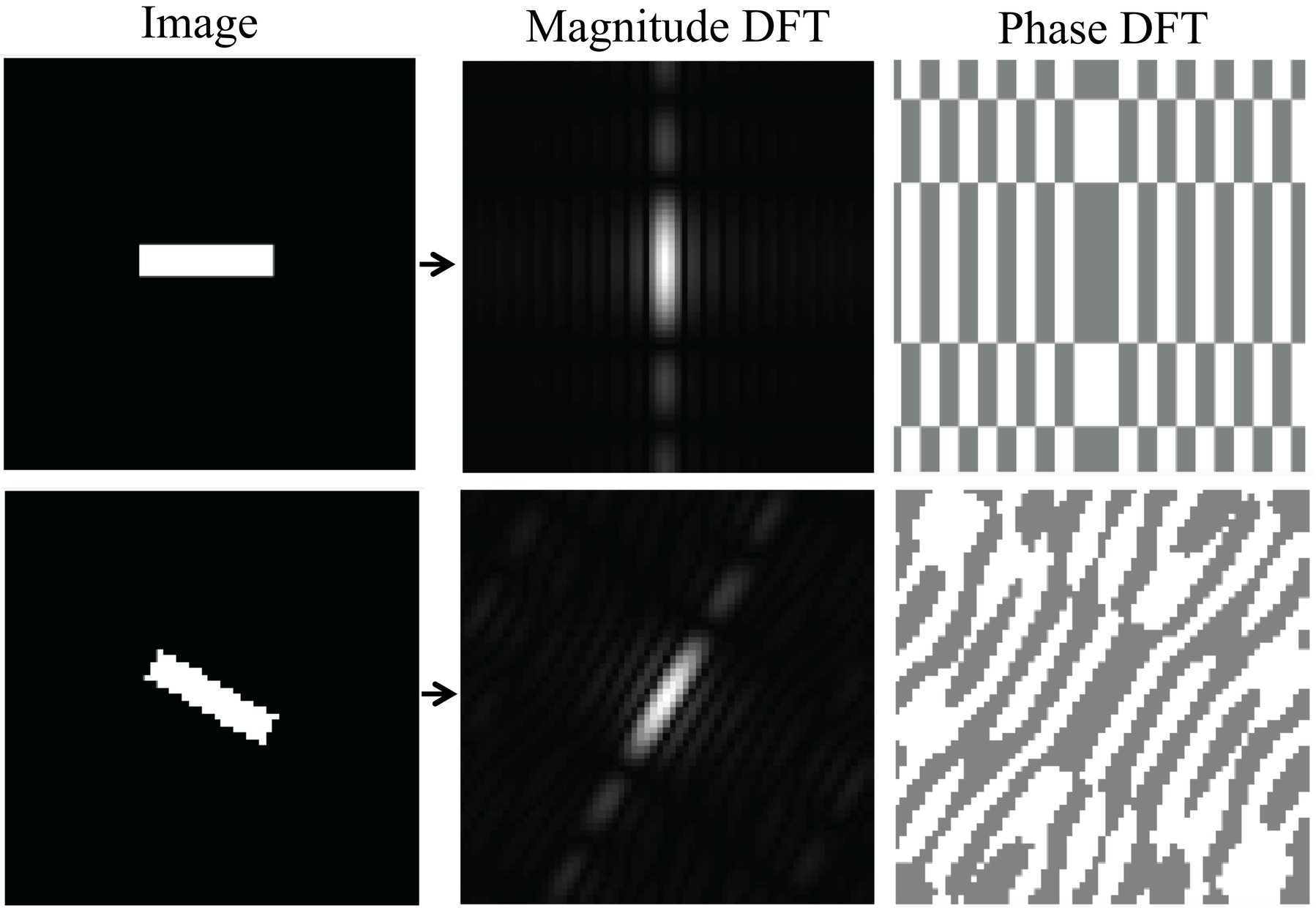
Some important Fourier transforms



Scale

Small image details produce content in high spatial frequencies

Some important Fourier transforms

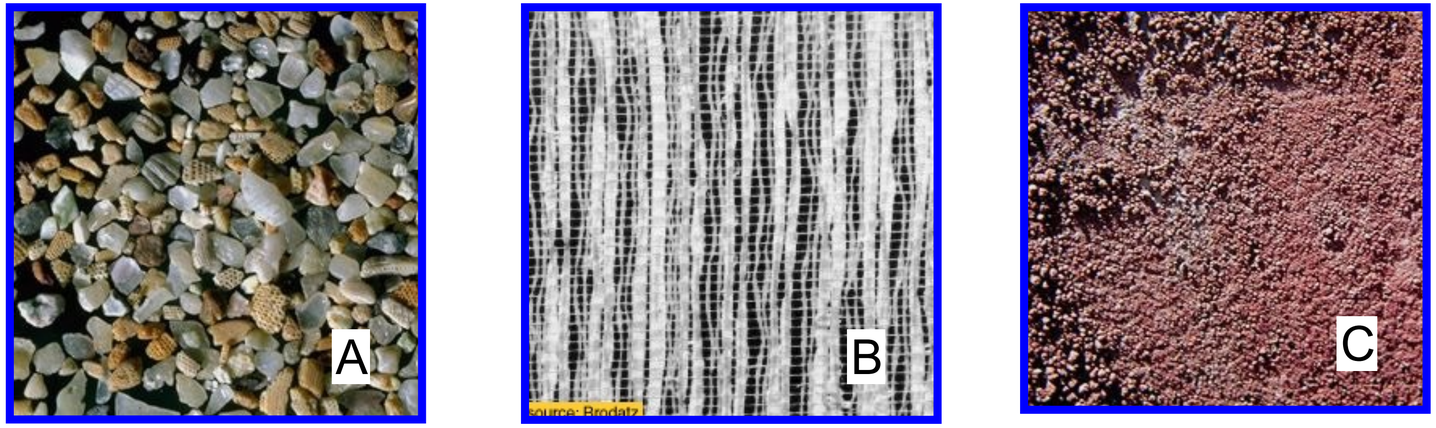


Orientation

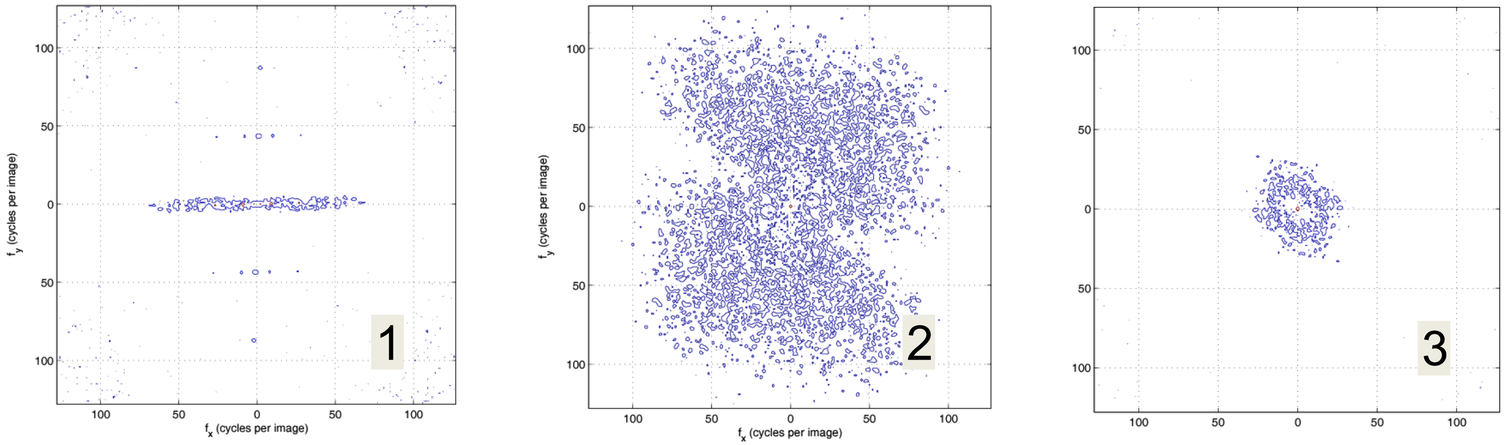
A line transforms to a line oriented perpendicularly to the first.

The DFT Game: find the right pairs

Images



DFT magnitude

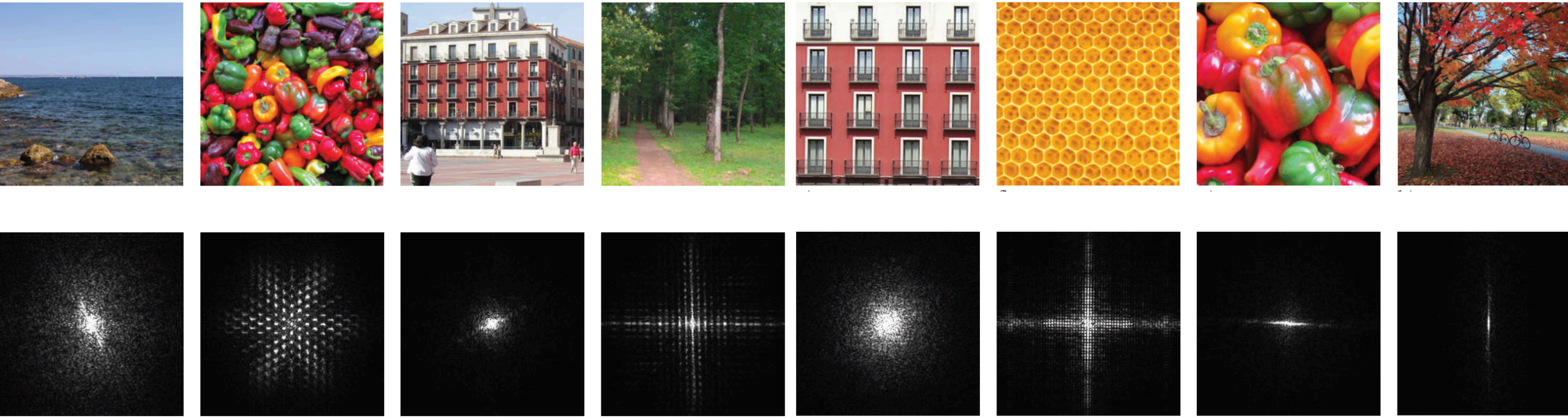


fx(cycles/image pixel size)

fx(cycles/image pixel size)

fx(cycles/image pixel size)

The DFT Game: find the right pairs



Solution in notes!

The inverse Discrete Fourier transform

2D Discrete Fourier Transform (DFT) transforms an image *f* [*n,m*] into *F* [*u,v*] as:

$$F[u,v] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f[n,m] \exp\left(-2\pi j \left(\frac{un}{N} + \frac{vm}{M}\right)\right)$$

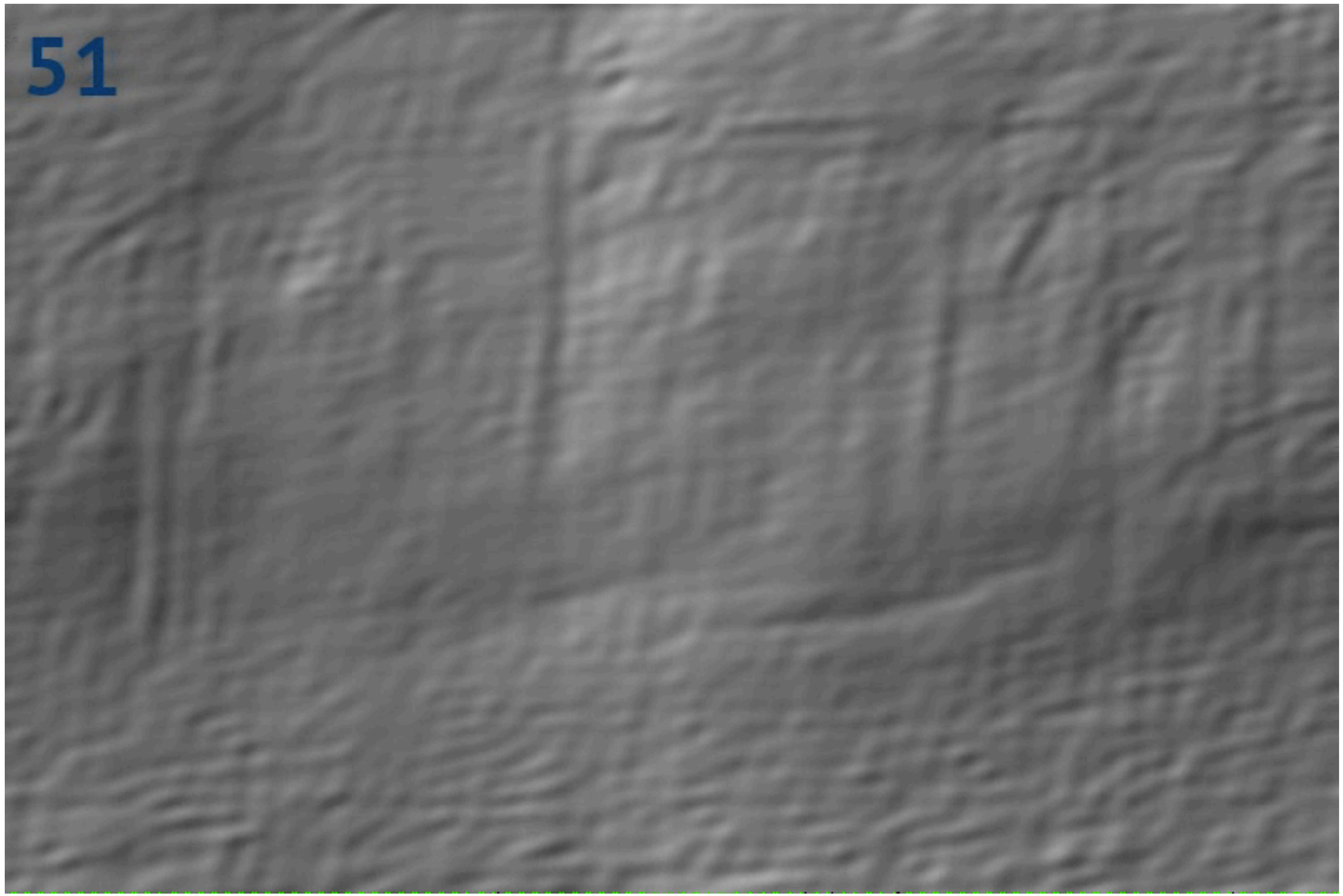
The inverse of the 2D DFT is:

$$f[n,m] = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} F[u,v] \exp\left(+2\pi j \left(\frac{un}{N} + \frac{vm}{M}\right)\right)$$

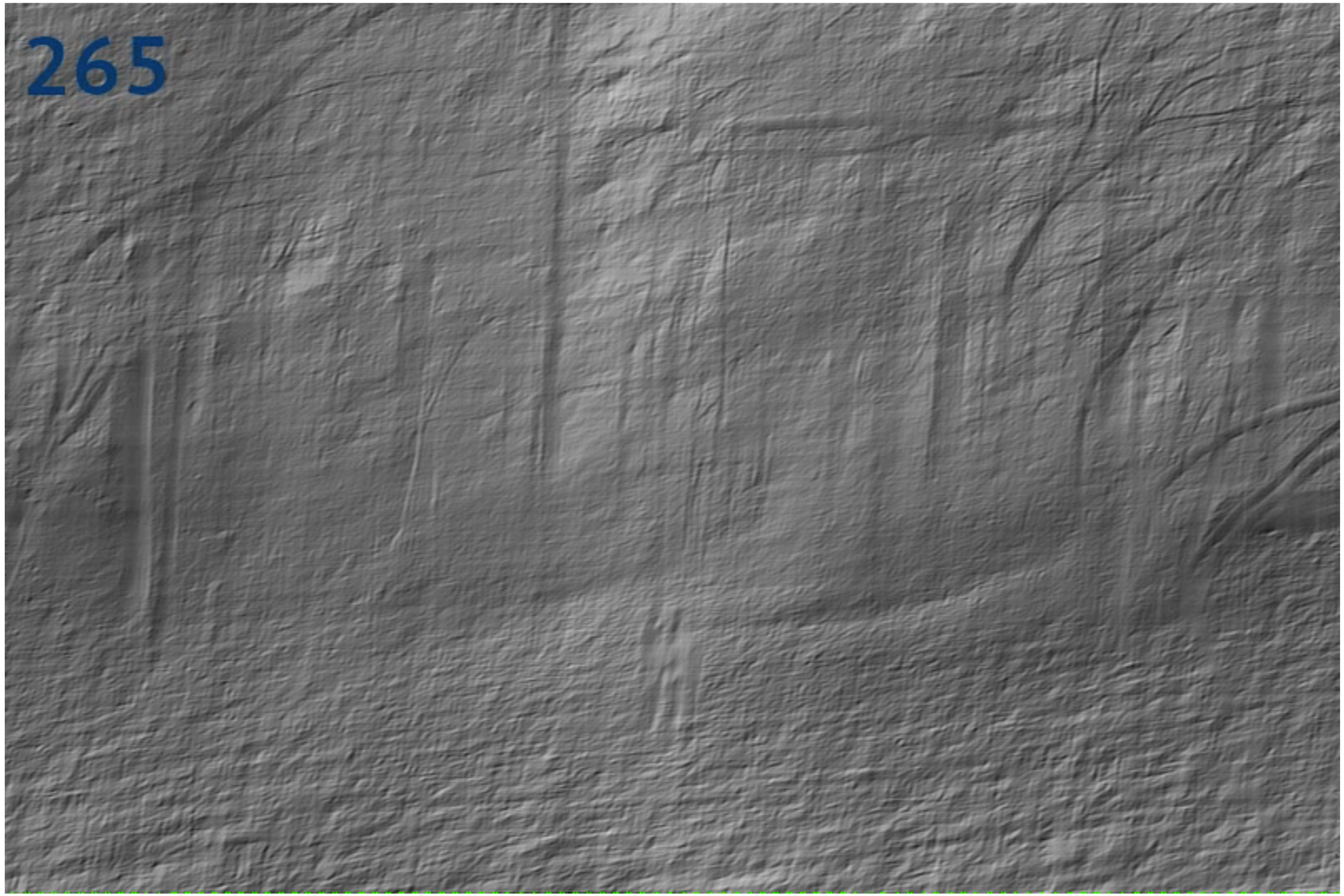
Reconstruct an image, low frequency to high

Reconstruct an image, low frequency to high

Reconstruct an image, low frequency to high



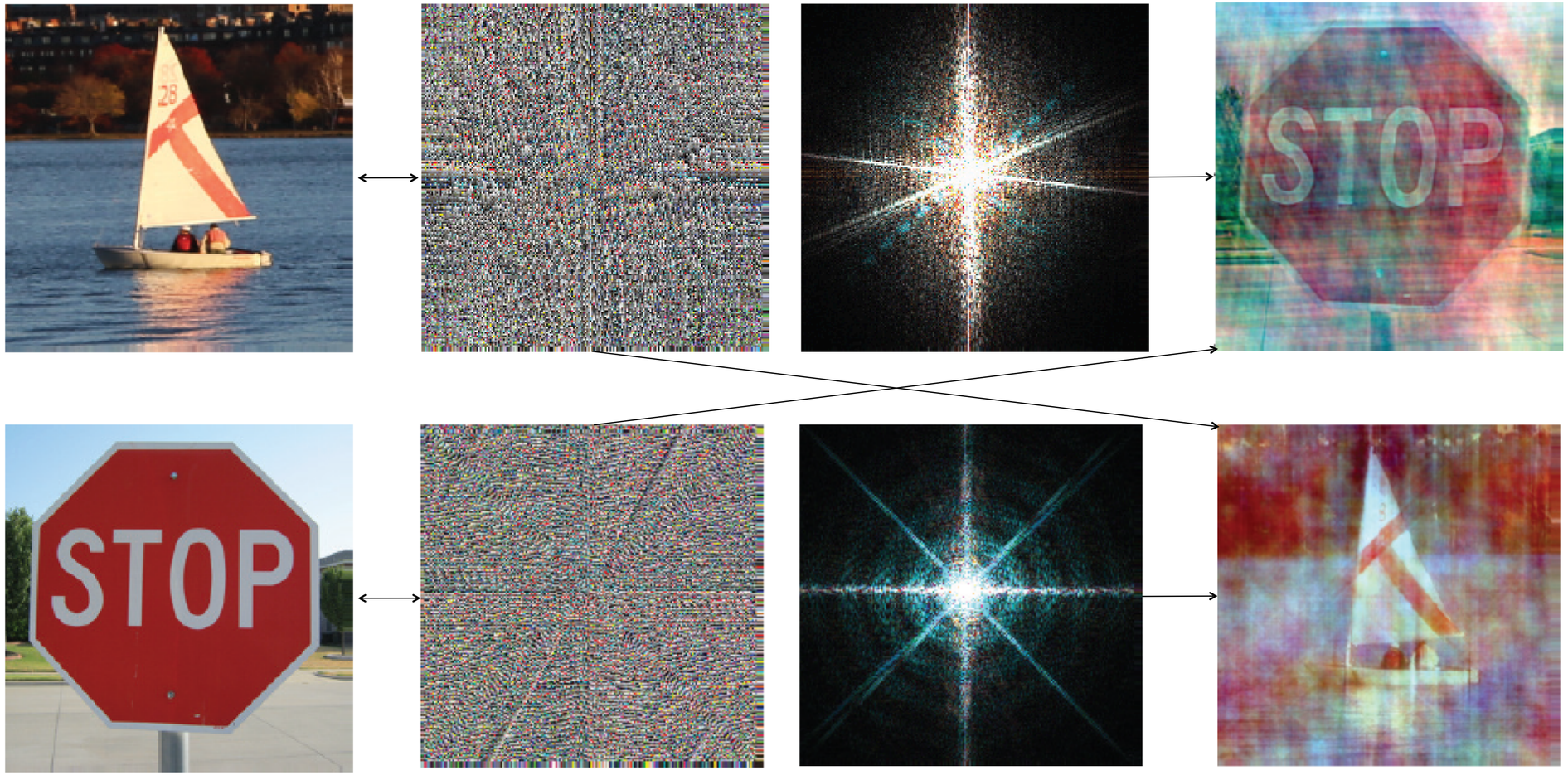
Reconstruct an image, low frequency to high

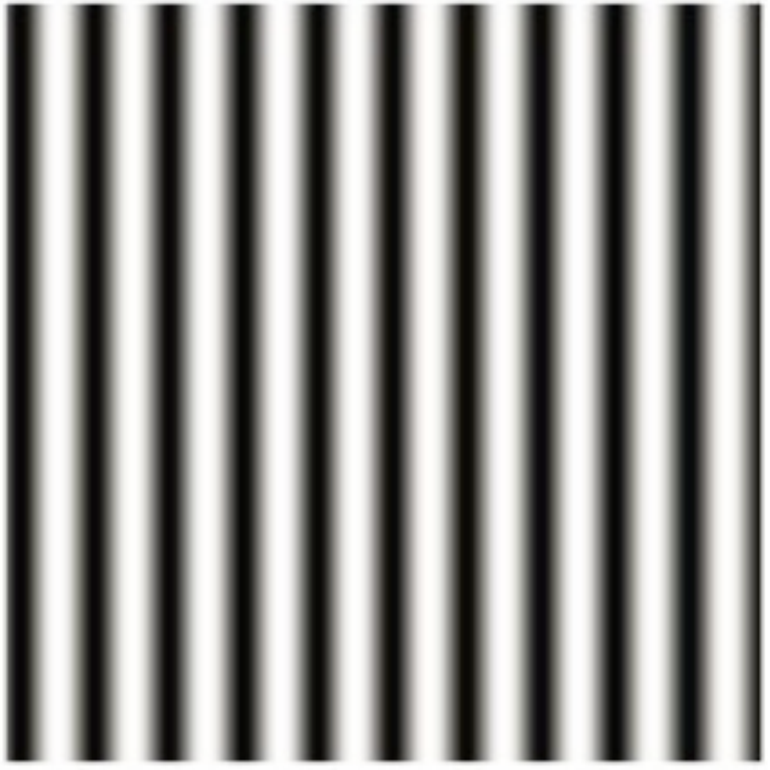


Reconstruct an image, low frequency to high

Phase and Magnitude

$$F[u,v] = A[u,v]\exp(j\theta[u,v])$$





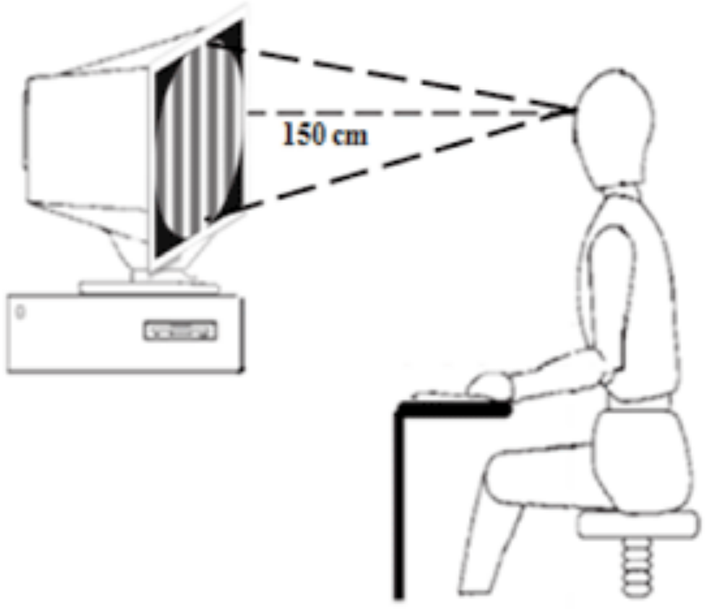


Figure 1. Stimulus presentation scheme. The stimuli were originally calibrated to be seen at a distance of 150 cm in a 19" display.

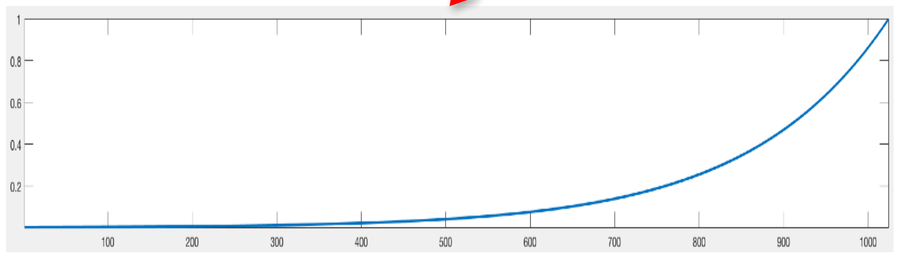
Campbell & Robson chart

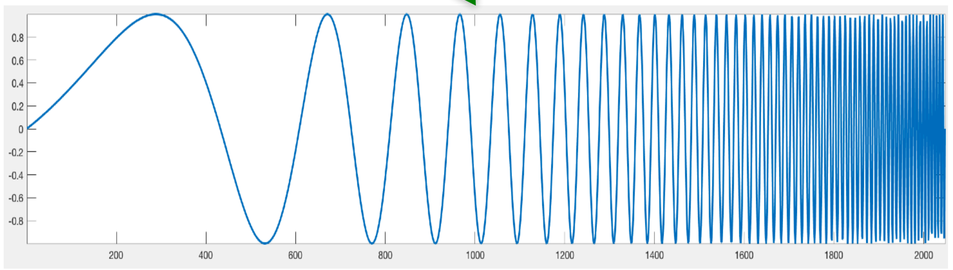
Let's define the following image:

$$\mathbf{I}[n,m] = A[n]\sin(2\pi f[m]\, m/M)$$

With:

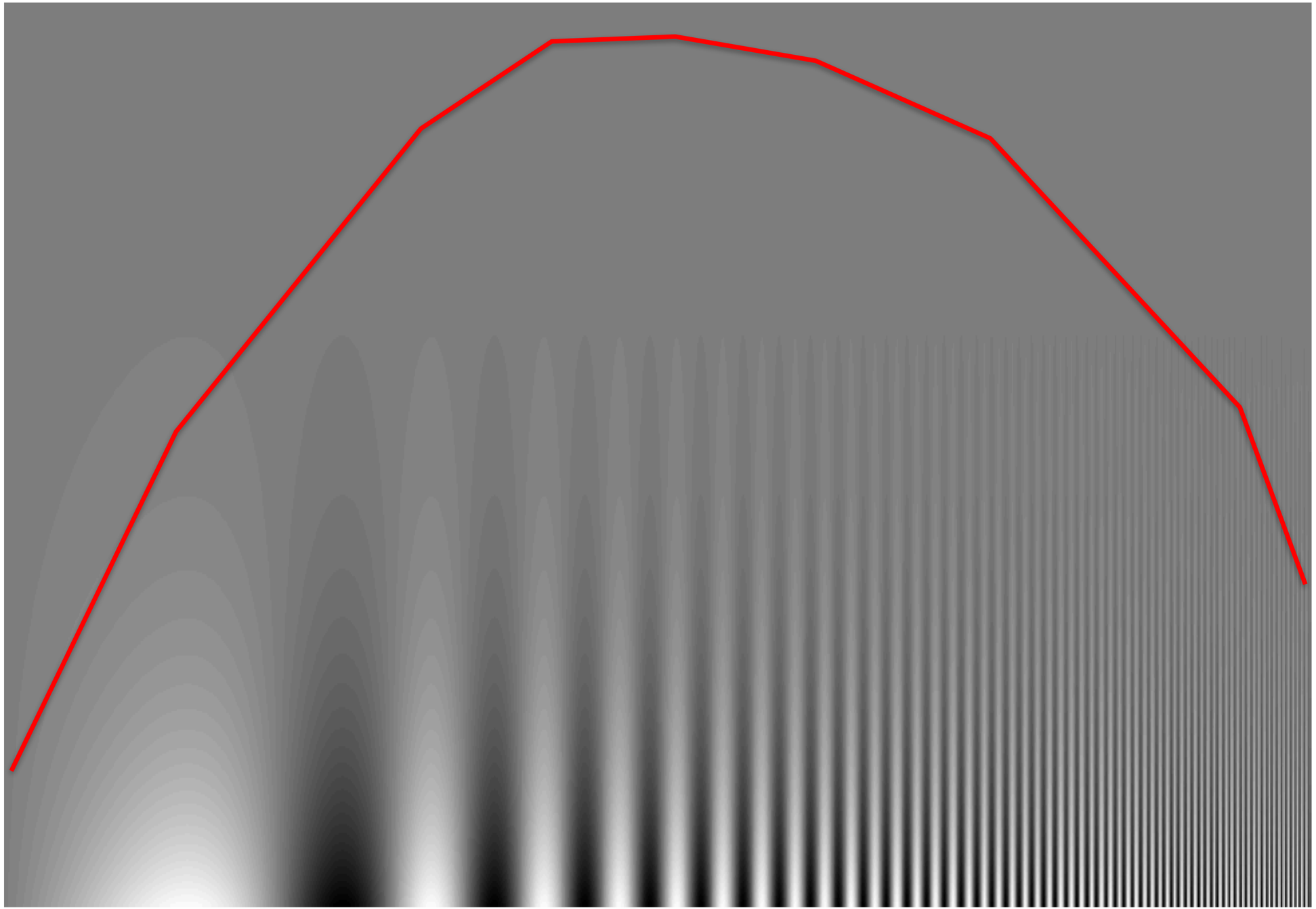
$$A[n] = A_{min}\left(\frac{A_{max}}{A_{min}}\right)^{n/N} \qquad f[m] = f_{min}\left(\frac{f_{max}}{f_{min}}\right)^{m/M}$$





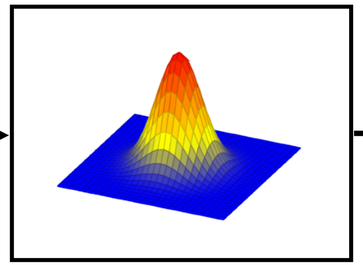
What do you think you should see when looking at this image?

$$\mathbf{I}[n,m] = A[n]\sin(2\pi f[m]\, m/M)$$

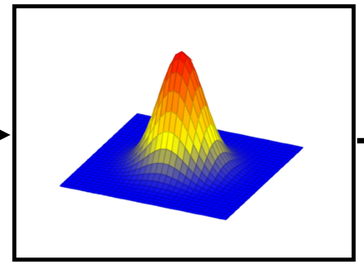


What about the opposite of blurring?

Gaussian filter

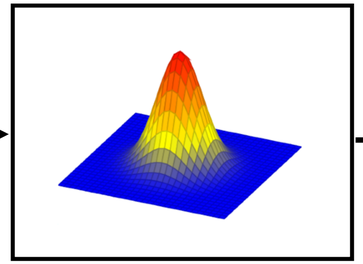


Laplacian filter



Laplacian filter

Gaussian filter



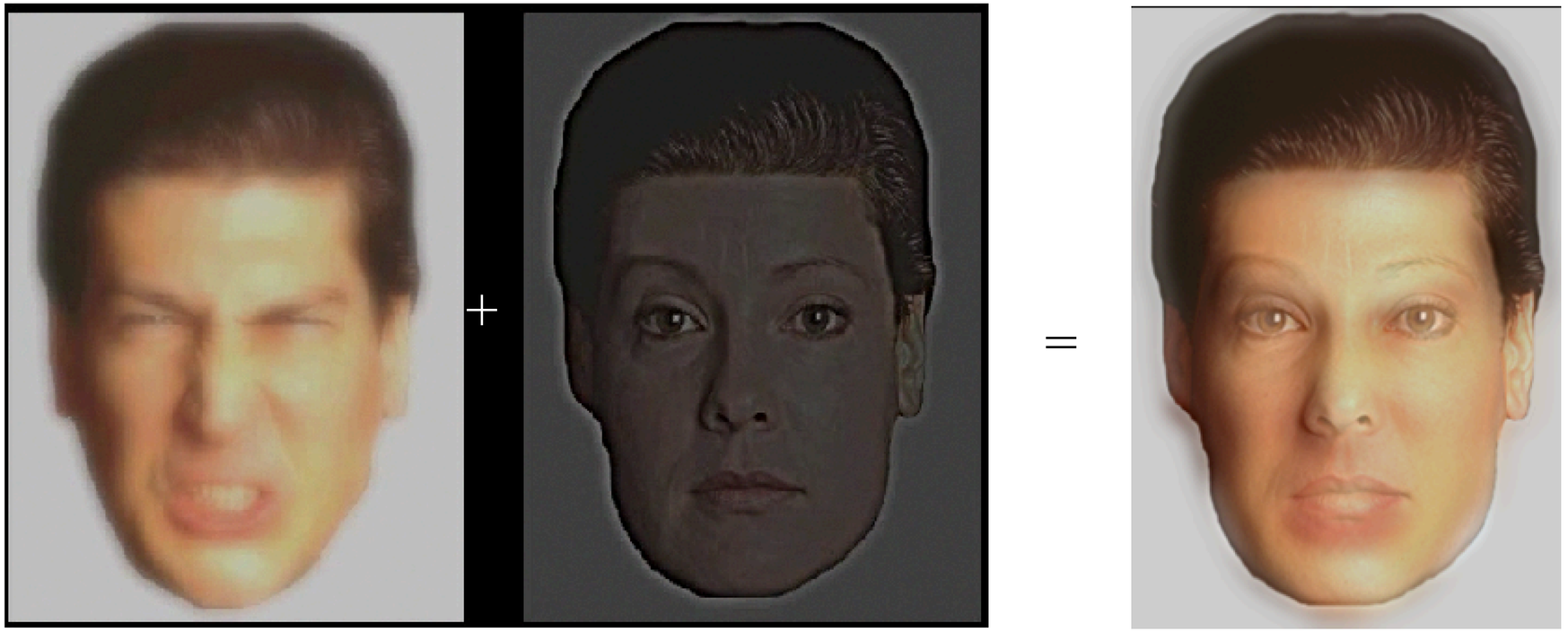
+

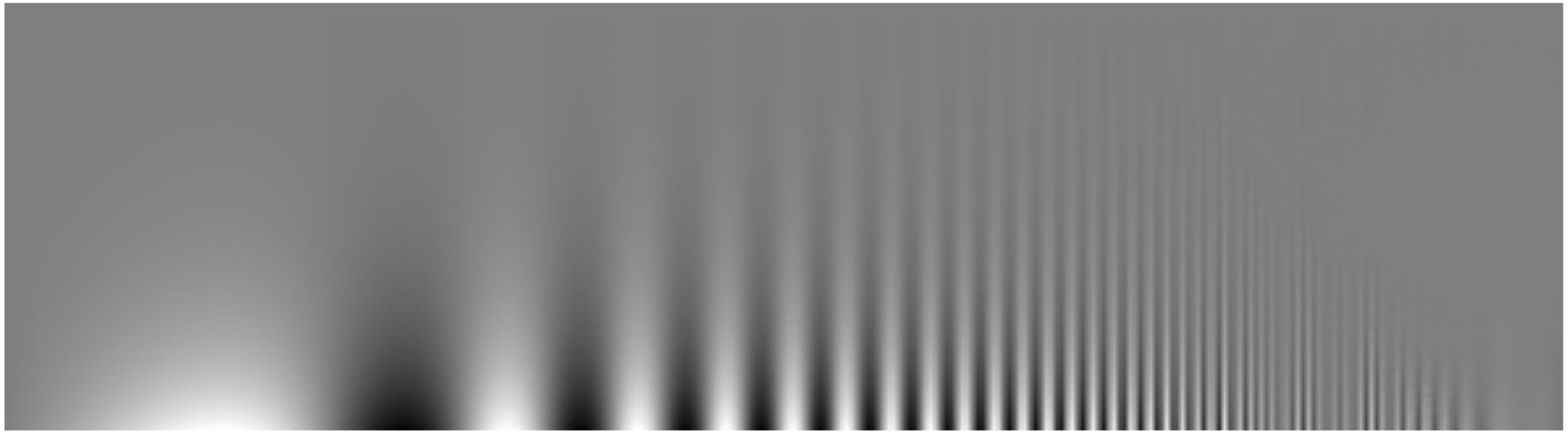
=

Hybrid Images

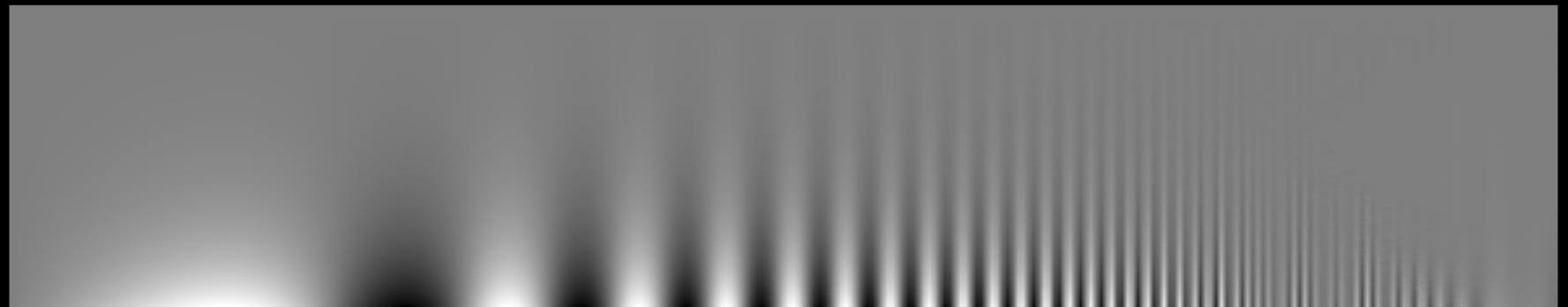
Oliva & Schyns

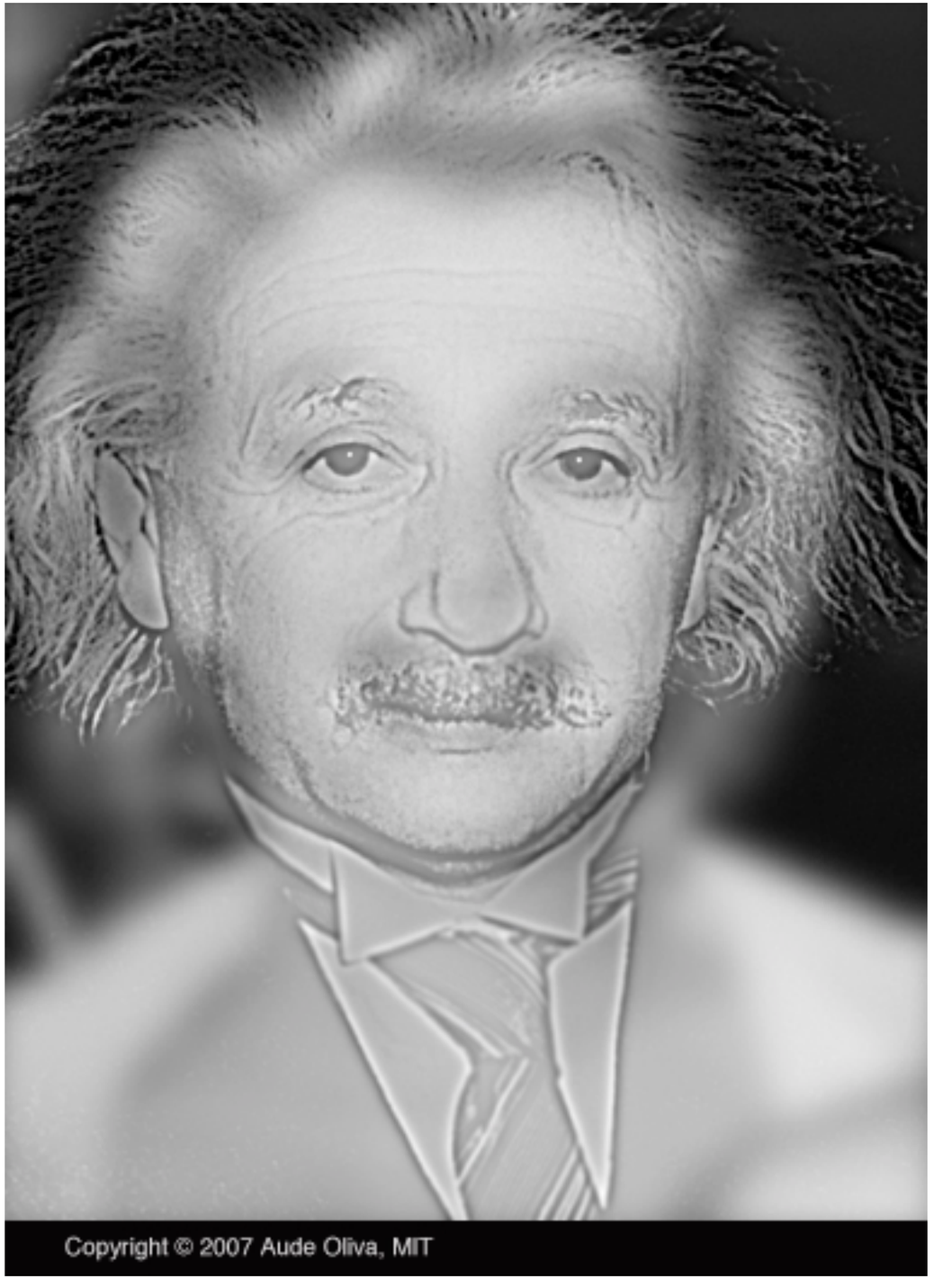
Hybrid Images

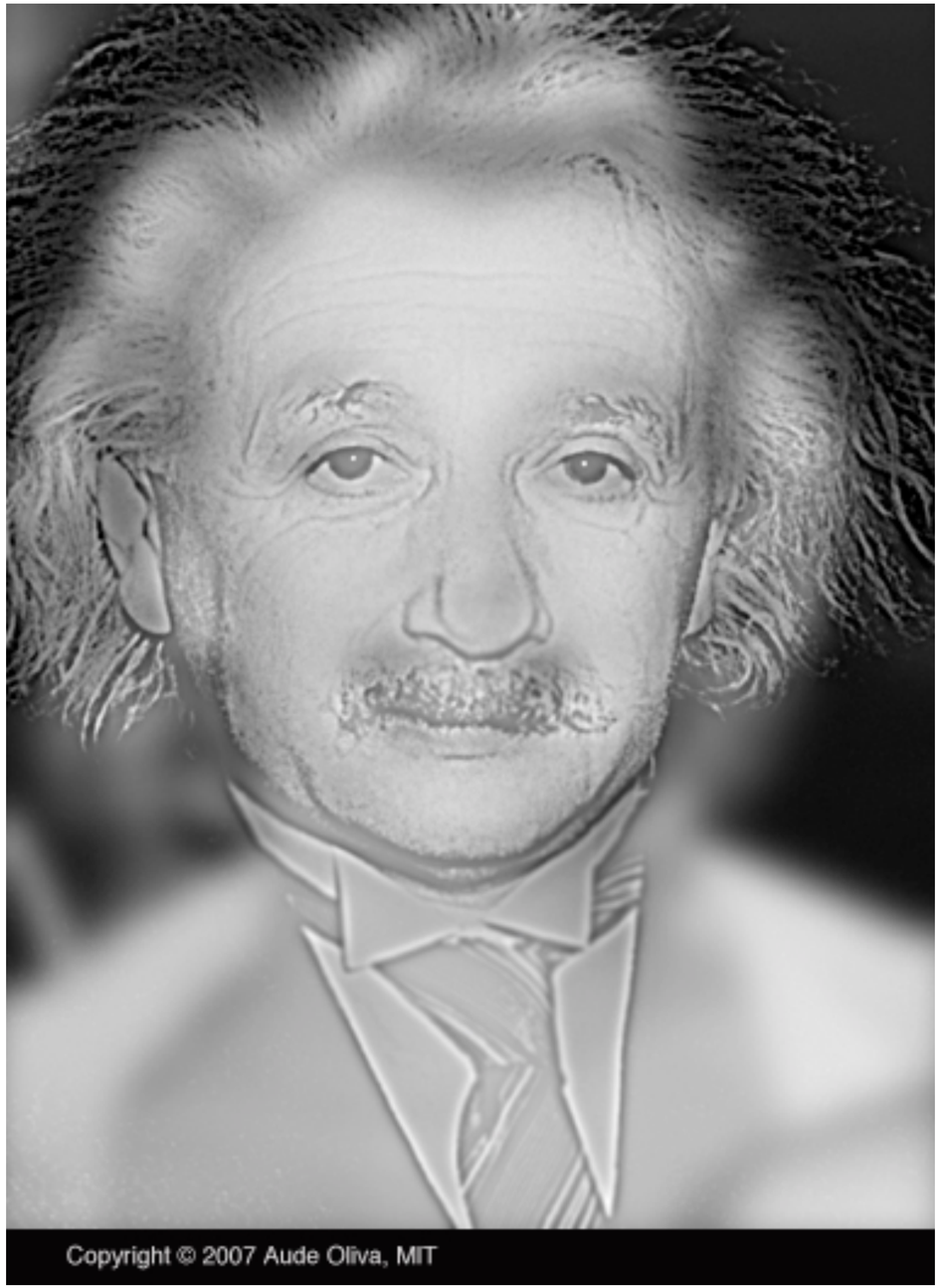


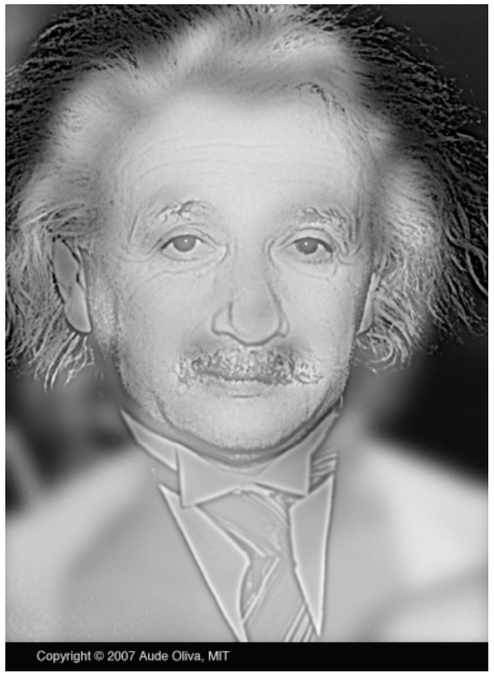


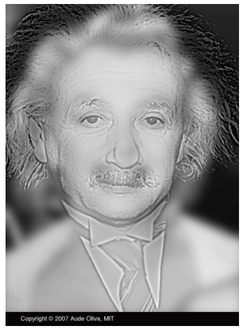
Hybrid Images




Copyright © 2007 Aude Oliva, MIT







http://cvcl.mit.edu/hybrid_gallery/gallery.html