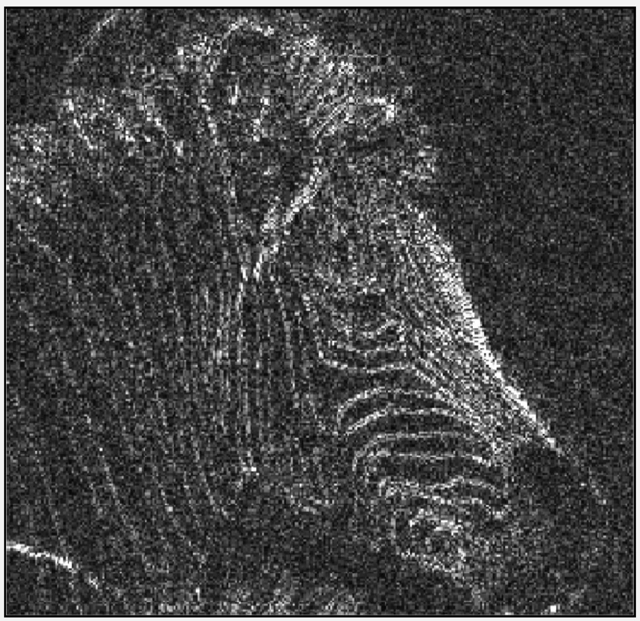
Department of Computer Science
University of Bristol

Image Processing and Computer Vision

www.ole.bris.ac.uk/bbcswebdav/courses/COMS30121_2018/content
www.ole.bris.ac.uk/bbcswebdav/courses/COMSM0020_2018/content



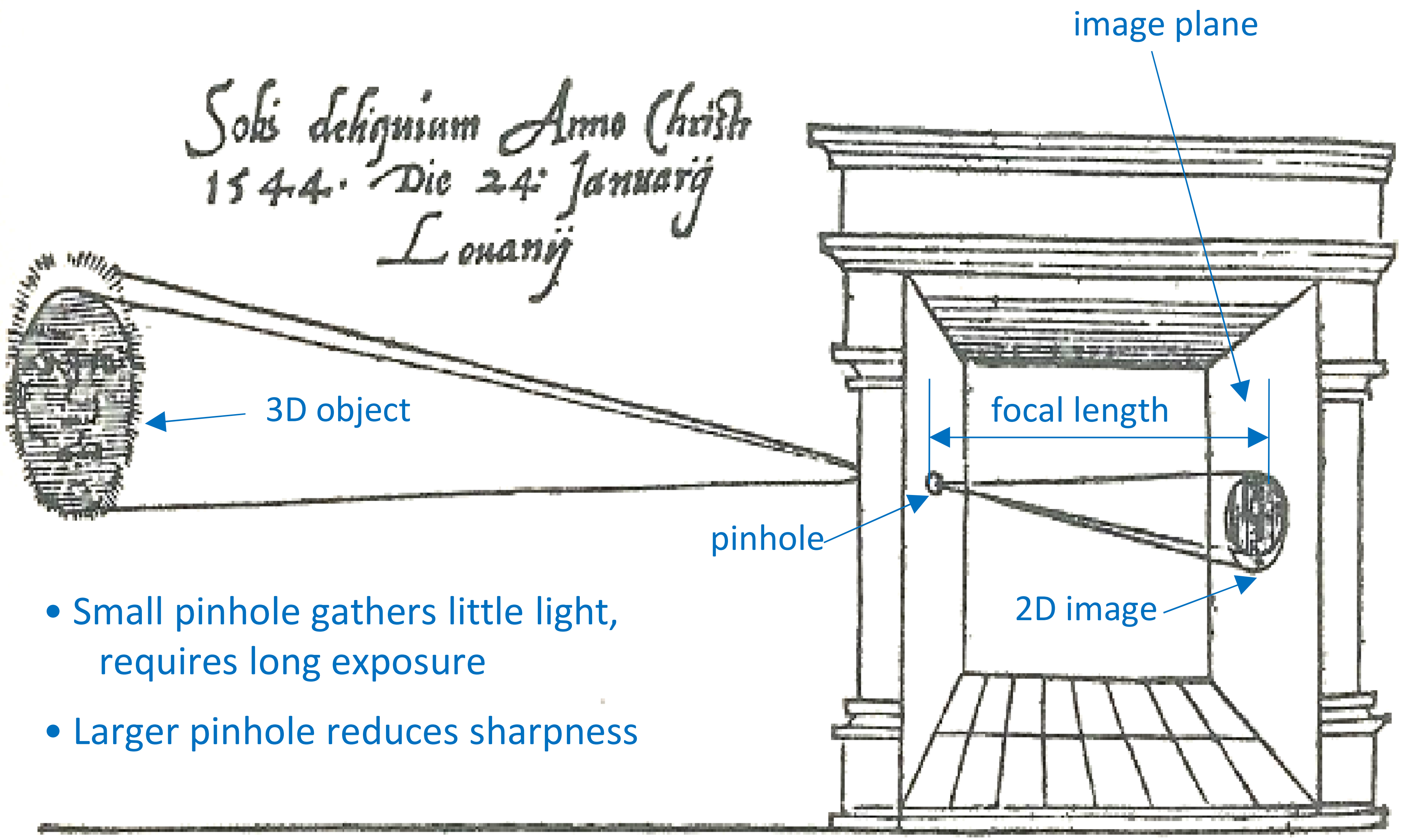
Lecture 03

Frequency Domain & Image Transforms

Andrew Calway | Tilo Burghardt | Sion Hannuna

24 Slides

Recap: The Camera Obscura (Pinhole Camera)



- Small pinhole gathers little light, requires long exposure
- Larger pinhole reduces sharpness

First published picture of camera obscura in Gemma Frisius' 1545 book *De Radio Astronomica et Geometrica*

Recap: OpenCV(C++) Image Representation

$$f(x_1, x_2, \ldots, x_m) = (c_1, c_2, \ldots, c_n)$$

$$f : \Re^m \to \Re^n$$

draw.cpp

```
#include [...]
using namespace cv;

int main() {
  //create a red 256x256, 8bit, 3channel BGR image in a matrix container
  Mat image(256, 256, CV_8UC3, Scalar(0, 0, 255));
```

width & height

8 bits per channel

unsigned

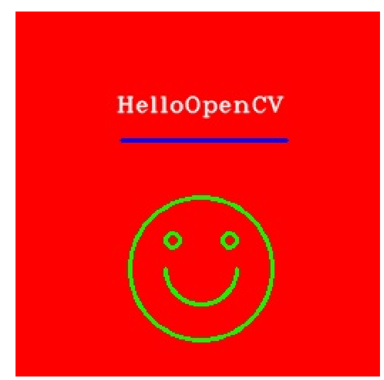
3 channels

blue

green

red

```
  //put white text HelloOpenCV
  putText(image, "HelloOpenCV", Point(70, 70),
    FONT_HERSHEY_COMPLEX_SMALL, 0.8, cvScalar(255, 255, 255), 1, CV_AA);
  //draw blue line under text
  line(image, Point(74, 90), Point(190, 90), cvScalar(255, 0, 0),2);
  //draw a green smile
  ellipse(image, Point(130, 180), Size(25,25), 180, 180, 360,
    cvScalar(0, 255, 0), 2);
  circle(image, Point(130, 180), 50, cvScalar(0, 255, 0), 2);
  circle(image, Point(110, 160), 5, cvScalar(0, 255, 0), 2);
  circle(image, Point(150, 160), 5, cvScalar(0, 255, 0), 2);
  //save image to file
  imwrite("myimage.jpg", image);
  //free memory occupied by image
  image.release();
  return 0;
}
```

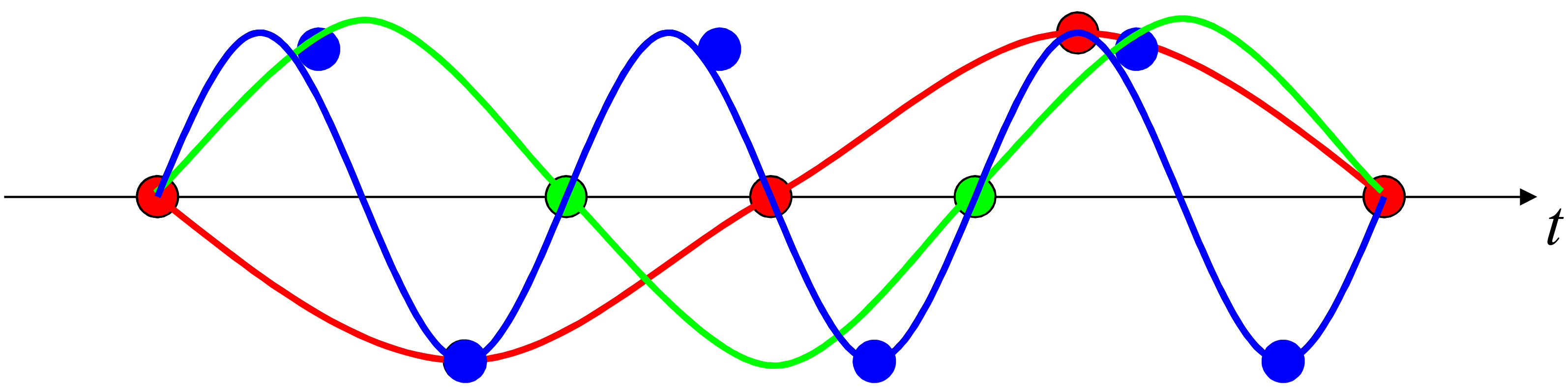


Recap: Shannon's Sampling Theorem

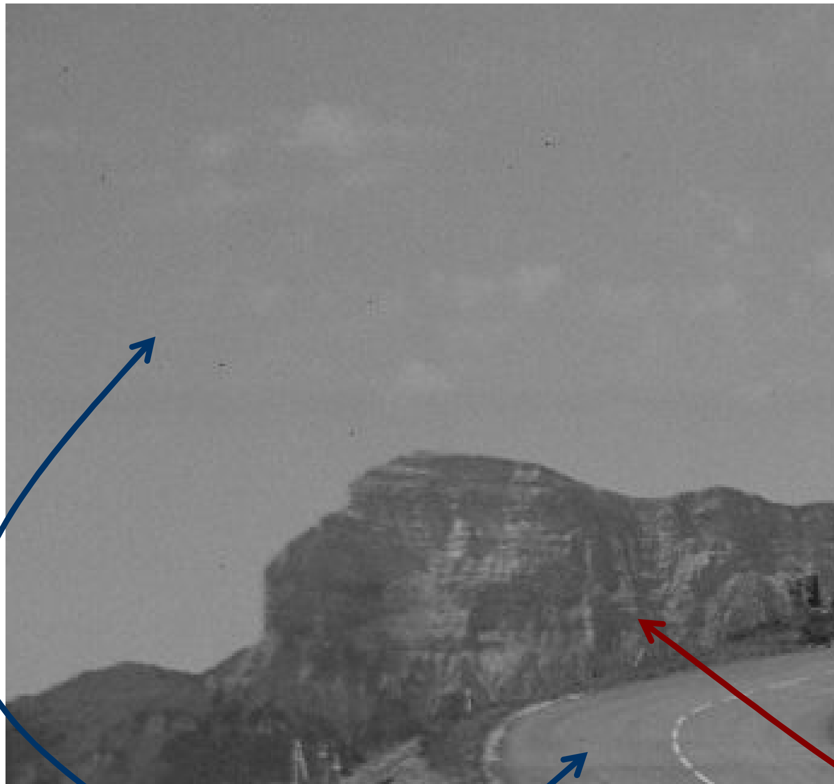
"An analogue signal containing components up to some maximum frequency ***u*** may be completely reconstructed by regularly spread samples, provided the sampling rate is above 2***u*** samples per second."

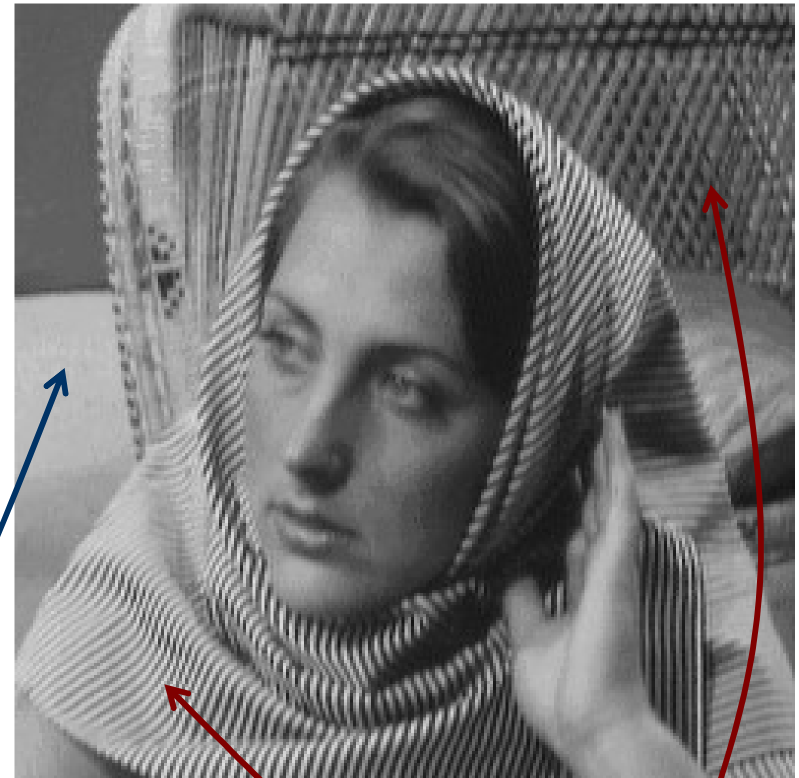
Claude Shannon

Also referred to as the Shannon-Nyquist criterion: Sampling **must** be performed **above twice** the highest (spatial) frequency of the signal to be lossless.



Recap: Intuition of Spatial Frequency





Slowly changing → low frequency

Rapidly changing →high frequency

Jean-Baptiste Joseph Fourier

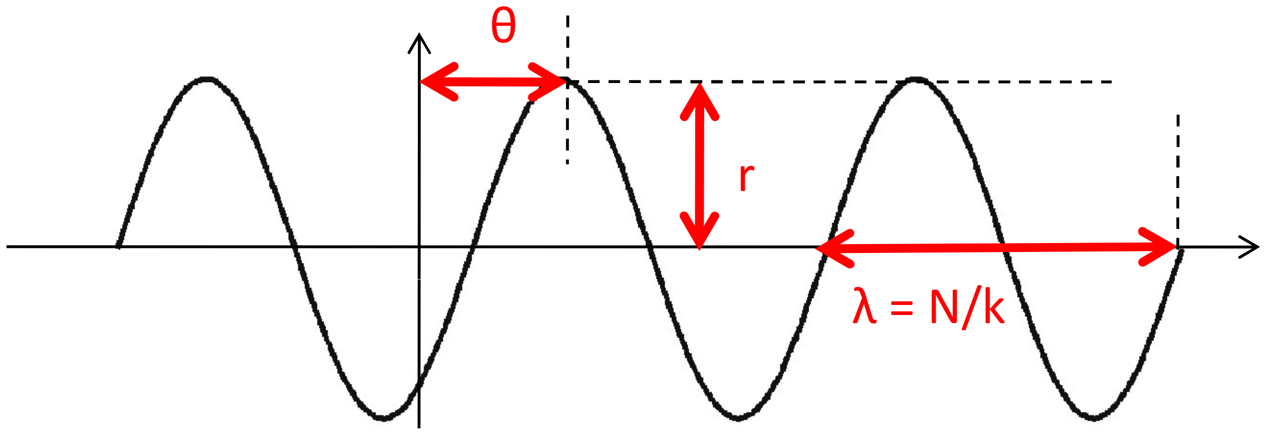
Fourier's Theorem

$$f(x) = \int a_n \cos(nx) + b_n \sin(nx) \quad \delta n$$

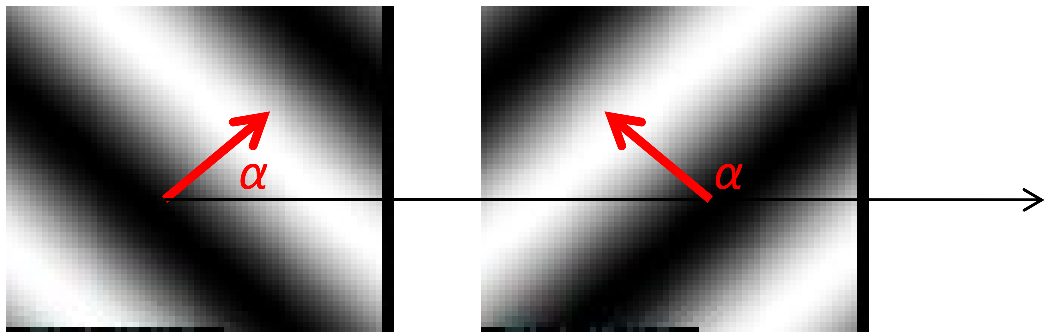
Animation by Lucas V Barbosa

Representing 2D Basis Functions

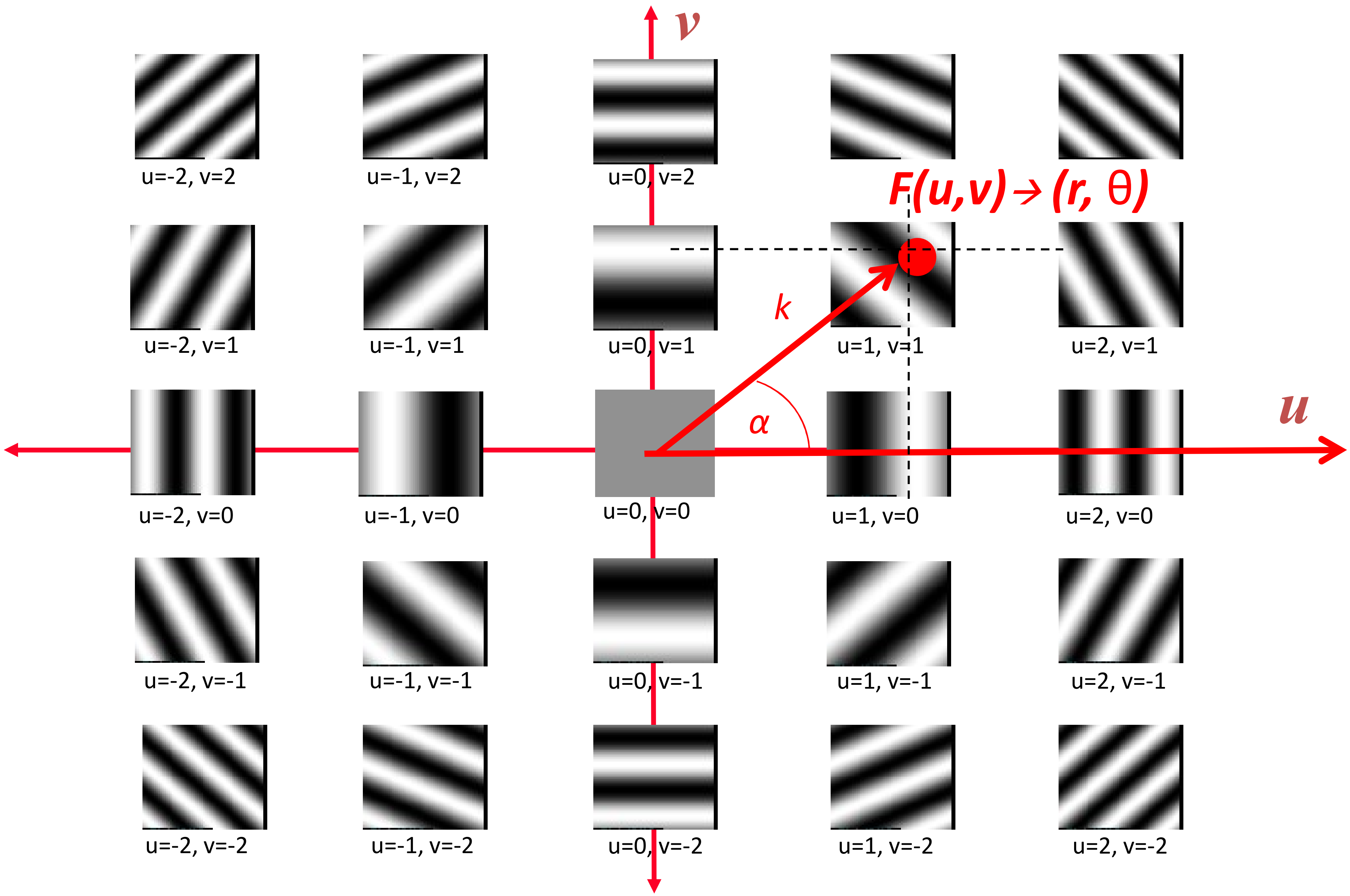
- we can specify any 1-dimensional sine-like wave by three parameters: the frequency *k* (cycles per second), the start phase θ (shift in degrees), the amplitude *r* (peak value)



- for a 2-dimensional wave we add as a parameter the direction encoded by an angle *α* (rotation of wavefront in degrees)



‘Fabric’ of the 2D Fourier Space (as kernels)

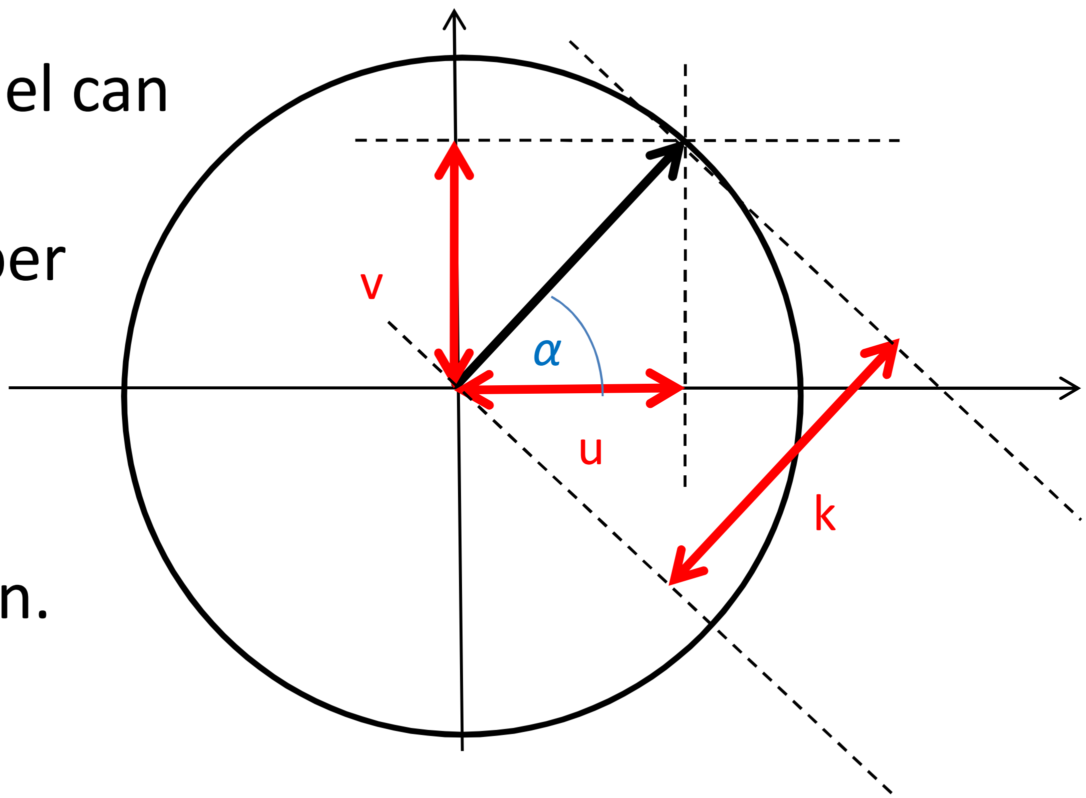


Euler's Equation

Instead of representing kernel functions as explicit sine or cosine functions we can campactly represent them by exponentials:

$$e^{2\pi i(ux+vy)/N} = \cos(2\pi(ux+vy)/N) + i\sin(2\pi(ux+vy)/N)$$

Thus, every kernel can be indexed by a complex number (k, α) in polar coordinates or (u,v) in standard complex notation.



Change of Base: The Fourier Transform

Each term of the Fourier Transform (FT) is composed of the sum of all values of the image function *f(x,y)* multiplied by a particular kernel at a particular frequeny and orientation specified by *(u,v)*:

$$F(u,v) = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} \underbrace{\boxed{f(x,y)}}_{\text{image}} \; \underbrace{\boxed{e^{(-2\pi i(ux+vy)/N)}}}_{\text{kernels (probing functions)}}$$

All kernels together form a new orthogonal basis for our image.

Thus, we have transformed the image *f* from a spatial domain indexed in *(x,y)* to a frequency domain representation in *(u,v)*.

Components of the Frequency Domain

- *F(u,v)* is a complex number and has real and imaginary parts:

$$F(u,v) = R(u,v) + iI(u,v)$$

- Magnitudes (forming the Power Spectrum):

$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$

- Phase Angles (forming the Phase Spectrum):

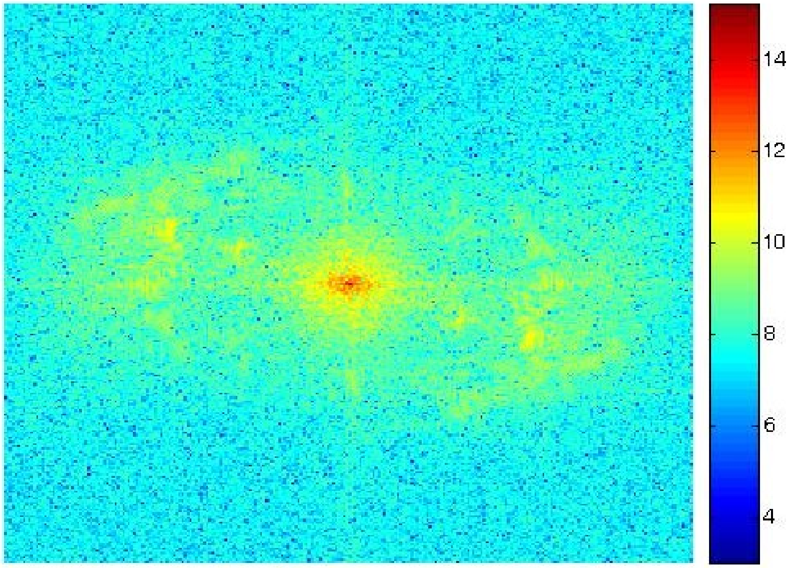
$$\theta(u,v) = \tan^{-1}\left[I(u,v)/R(u,v)\right]$$

- Expressing *F(u,v)* in polar coordinates ***(r,*** θ***)*** :

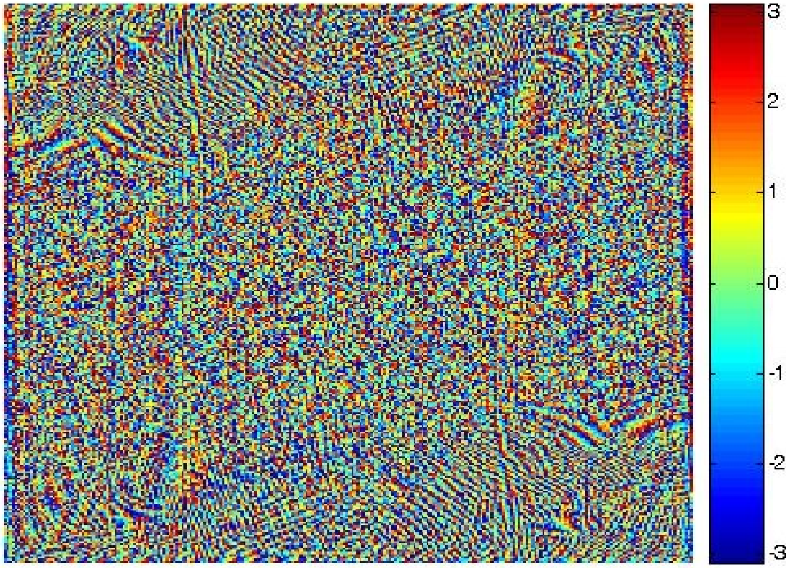
$$F(u,v) = |F(u,v)|e^{i\theta(u,v)} = re^{i\theta}$$

Example: Power Spectrum and Phase Spectrum

$$f(x,y)$$



$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$



$$\theta(u,v) = \tan^{-1}\left[I(u,v)/R(u,v)\right]$$

2D Fourier Transform Pair

Given a fourier transform of a discrete function of two variables:

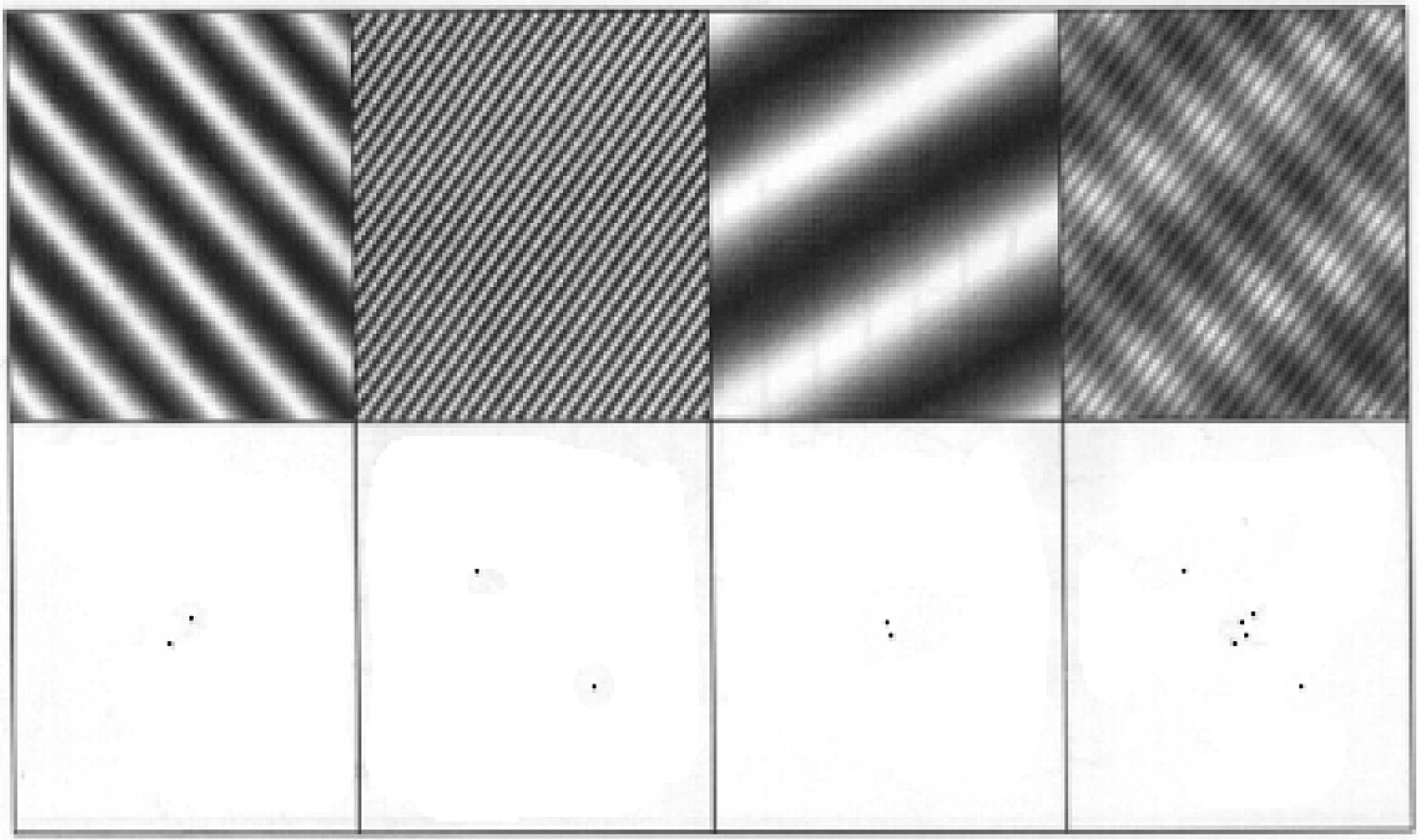
$$F(u,v) = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(x,y)\ e^{(-2\pi i(ux+vy)/N)}$$

There exists an inverse transform that can reconstruct the original image in spatial coordinates from its representation in the frequency domain. This is known as the Inverse Fourier Tansform:

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

Together the two equations form the Fourier Transform Pair.

Image Pairs: Spatial Domain vs Frequency Domain

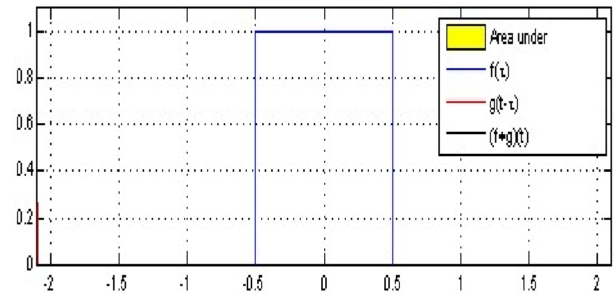


Recap: Convolution Operation

- ...quantifies the structural similarity of a kernel image ***h(x)*** as it is shifted over a target image ***f(x)***:

$$f * h = \int_{-\infty}^{+\infty} f(x-t)h(x)\,\partial t$$

Convolution symbol



Animation Source: Brian Amberg

- ... determines the effect of a system, i.e. the kernel ***h(x)***, on an input signal, i.e. ***f(x)***
- the result image is known as the \`response' of ***f*** to the kernel ***h***

Deriving the Convolution Theorem

Definition of Convolution

$$h(x) = f(x) * g(x) = \sum_{y} f(x-y)g(y)$$

Definition of Fourier Transform

$$H(u) = \sum_{x} \left(\sum_{y} f(x-y)g(y) \right) e^{(-iux2\pi/N)}$$

Reordering of Summations

$$H(u) = \sum_{y} g(y) \left(\sum_{x} f(x-y) e^{(-iux2\pi/N)} \right)$$

Substitution of $x=z+y$

$$H(u) = \sum_{y} g(y) \left(\sum_{z} f(z) e^{(-iu(z+y)2\pi/N)} \right)$$

Splitting of Exponential

$$H(u) = \sum_{y} g(y) e^{(-iuy2\pi/N)} \left(\sum_{z} f(z) e^{(-iuz2\pi/N)} \right)$$

Definition of Fourier Transform

$$H(u) = G(u) \cdot F(u)$$

Convolution in the Spatial/Frequency Domain

- Convolution Theorem:

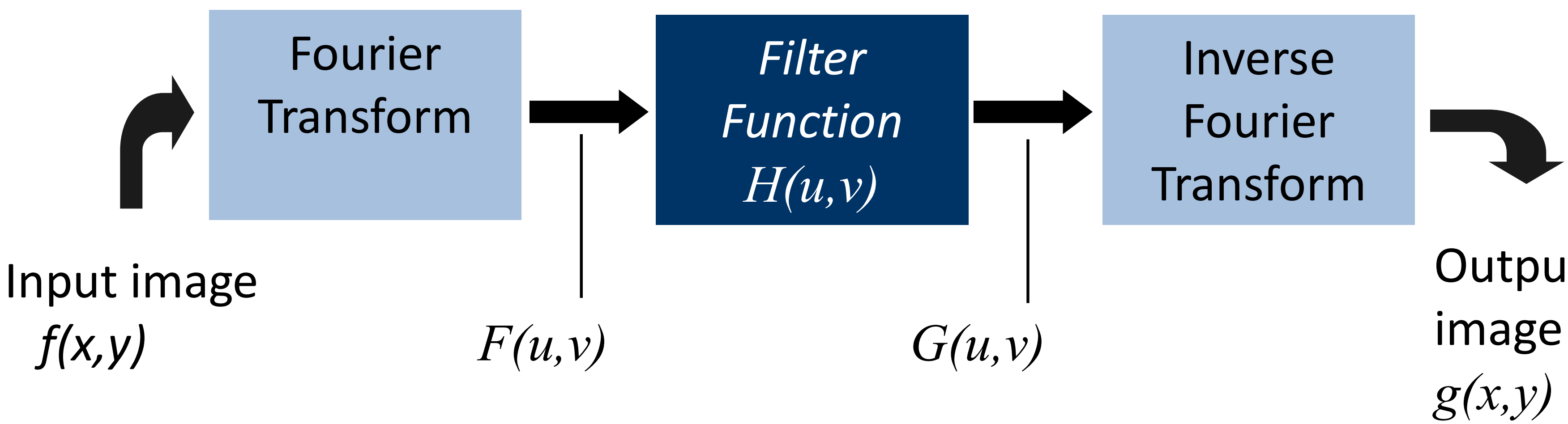
Convolution in spatial domain
is equivalent to
multiplication in frequency domain
(and vice versa)

$$g = f * h \quad \text{implies} \quad G = FH$$

$$g = fh \quad \text{implies} \quad G = F * H$$

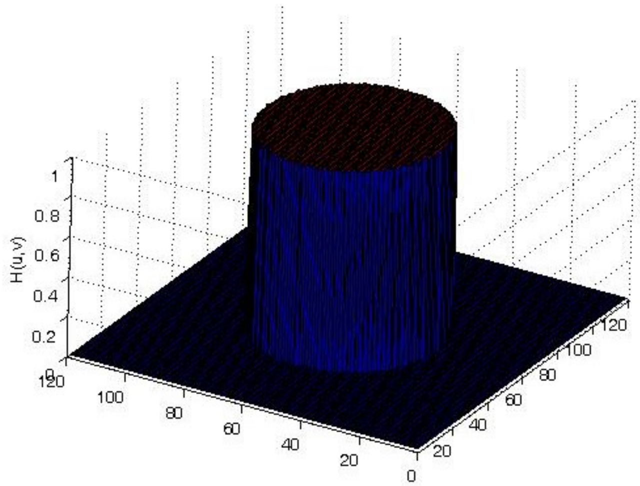
Fast Filtering using the Convolution Theorem

$$1\text{D}: \; G(u) = F(u)H(u) \qquad\qquad 2\text{D}: \; G(u,v) = F(u,v)H(u,v)$$



Low Pass Filtering

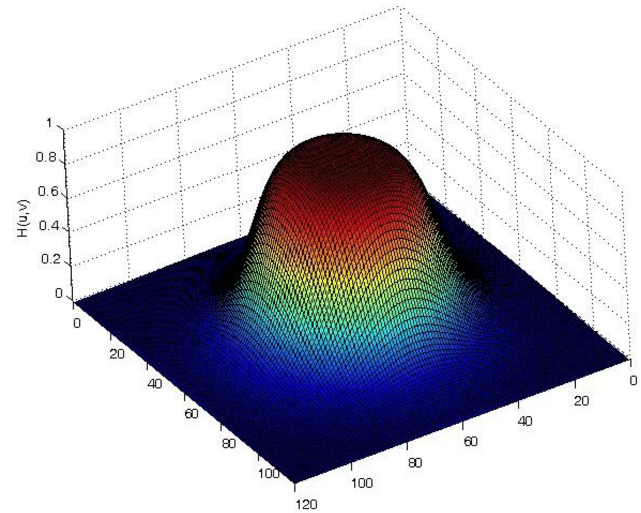
- 1D: turning the "treble" down on audio equipment!
- 2D: smooth image



$$H(u,v) = \begin{cases} 1 & r(u,v) \le r_0 \\ 0 & r(u,v) > r_0 \end{cases}$$

$$r(u,v) = \sqrt{u^2 + v^2}, \quad r_0 \text{ is the filter radius}$$

Butterworth's Low Pass Filter

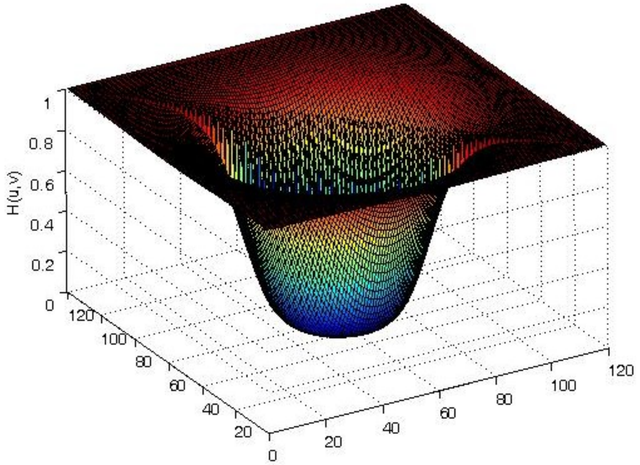


$$H(u,v) = \begin{cases} 1 & r(u,v) \le r_0 \\ 0 & r(u,v) > r_0 \end{cases}$$

$r(u,v) = \sqrt{u^2 + v^2}$, r_0 is the filter radius

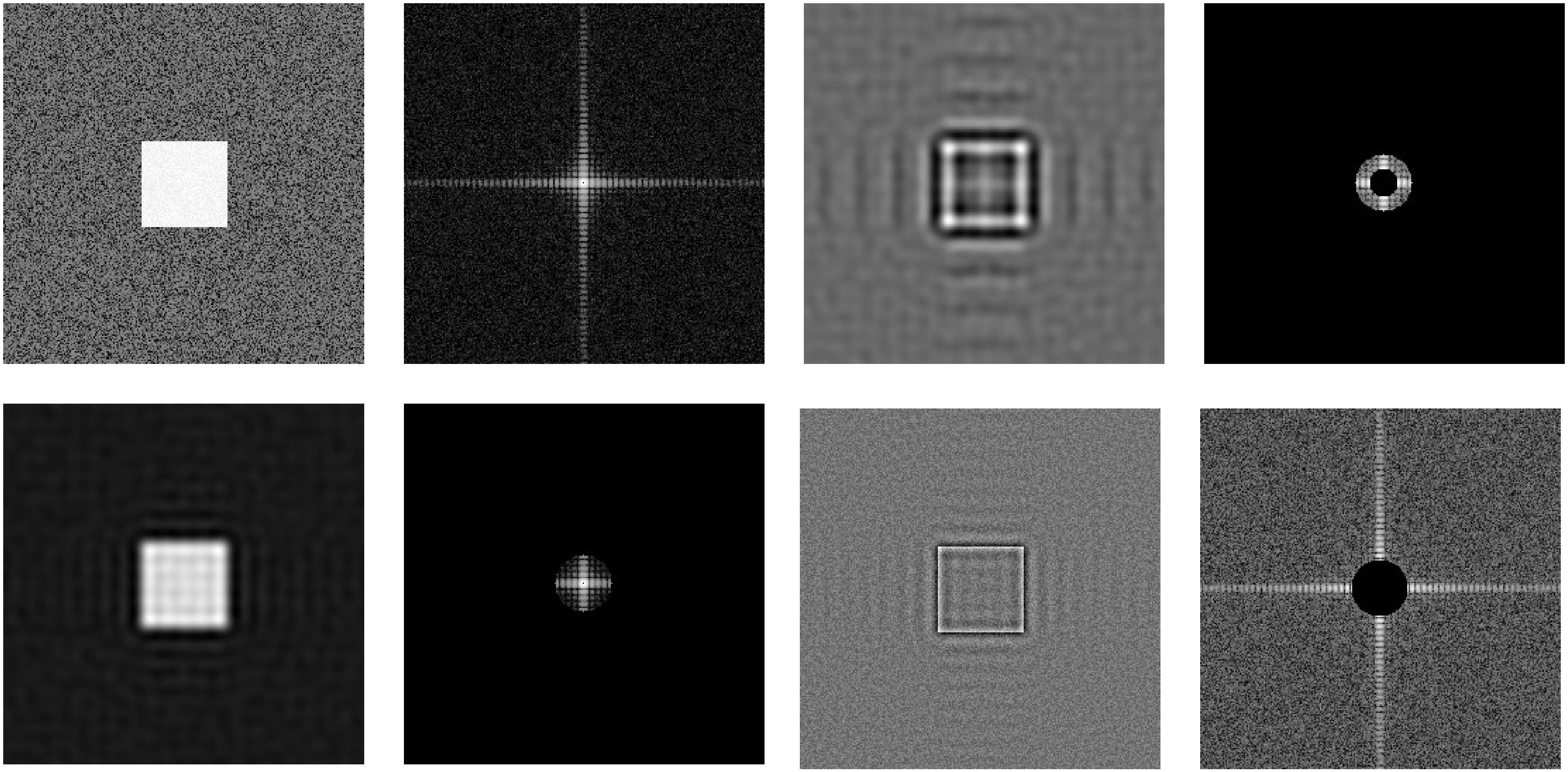
Butterworth's High Pass Filter

- 1D: turning the bass down on audio equipment!
- 2D: sharpen image



Order of $n=3$

$$H(u,v) = \frac{1}{1+[r_0 / r(u,v)]^{2n}} \quad \text{of order } n$$



Optional Excurse: Wiener De-Convolution

Norbert Wiener

- **<u>Idea:</u>** Restore an image by convolution with an adjusted inverse kernel that estimates the loss of information per frequency.

inverse of original kernel

$$\boxed{\frac{1}{H(f)}}\left[\frac{|H(f)|^2}{|H(f)|^2+\boxed{\frac{1}{\mathrm{SNR}(f)}}}\right]$$

estimated loss at frequency f