قسم الهندسة الحيوية
الطبية والمنظومات

جامعة القاهرة
كلية الهندسة

Computer Vision
404 B
Tutorial 2
18/02/20

Eng. Eman Marzban

Agenda:

Noise

Filtering

Canny Edge detector

Fourier transform

Agenda:

Noise

Filtering

Canny Edge detector

Fourier transform

Canny Edge detector

Algorithm CANNY_EDGE_DETECTOR

Given an image I:

1. apply CANNY_ENHANCER to I;
2. apply NONMAX_SUPPRESSION to the output of CANNY_ENHANCER;
3. apply HYSTERESIS_THRESH to the output of NONMAX_SUPPRESSION.

Emanuele Trucco, Alessandro Verri, "**Introductory Techniques for 3-D Computer Vision**"

http://www.cse.iitd.ernet.in/~pkalra/col783-2017/canny.pdf

Algorithm CANNY_ENHANCER

1. Apply Gaussian smoothing to I (algorithm LINEAR_FILTER of Chapter 3 with a Gaussian kernel discretising G), obtaining $J = I * G$.
2. For each pixel (i, j):

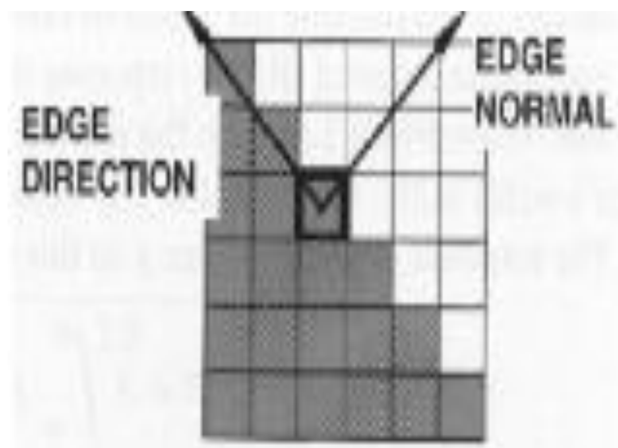
   (a) compute the gradient components, J_x and J_y (Appendix, section A.2);
   (b) estimate the edge strength

$$e_s(i, j) = \sqrt{J_x^2(i, j) + J_y^2(i, j)}$$

   (c) estimate the orientation of the edge normal

$$e_o(i, j) = \arctan \frac{J_y}{J_x}$$

The output is a *strength image*, E_s, formed by the values $e_s(i, j)$, and an *orientation image*, E_o, formed by the values $e_o(i, j)$.



1. Smoothing: Blurring of the image to remove noise.

(a) Original

(b) Smoothed

2. Finding gradients:

(a) Smoothed

(b) Gradient magnitudes

Algorithm NONMAX_SUPPRESSION

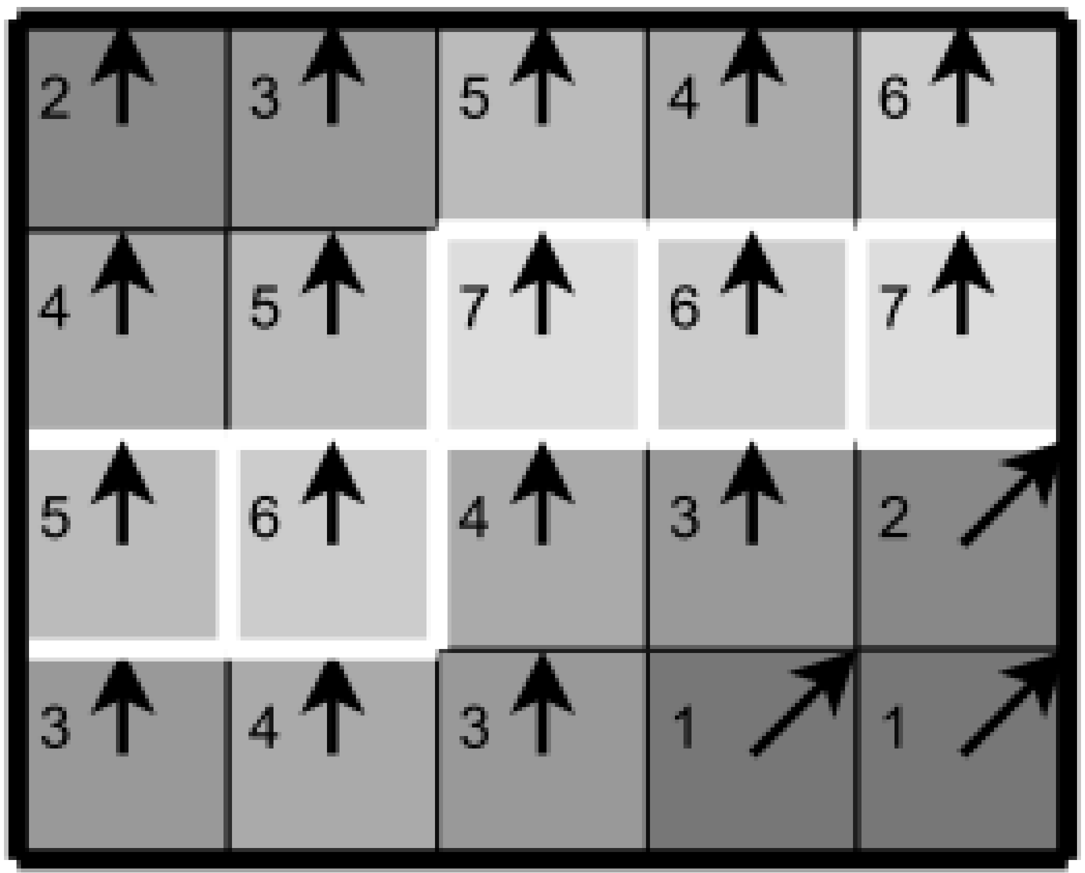
The input is the output of CANNY_ENHANCER, that is, the edge strength and orientation images, E_s and E_o. Consider the four directions $d_1 \dots d_4$, identified by the $0°$, $45°$, $90°$ and $135°$ orientations (with respect to the horizontal axis image reference frame).

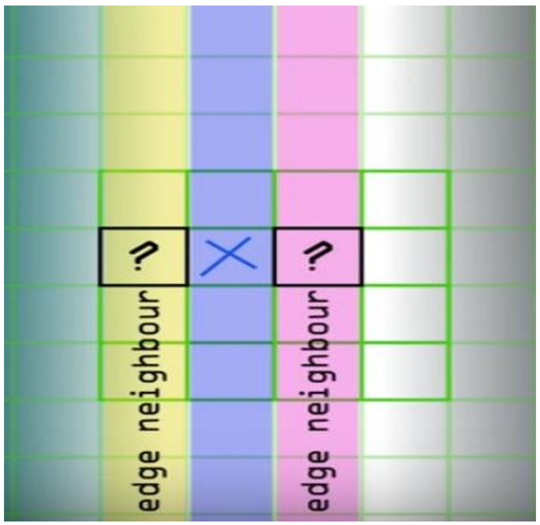
For each pixel (i, j):

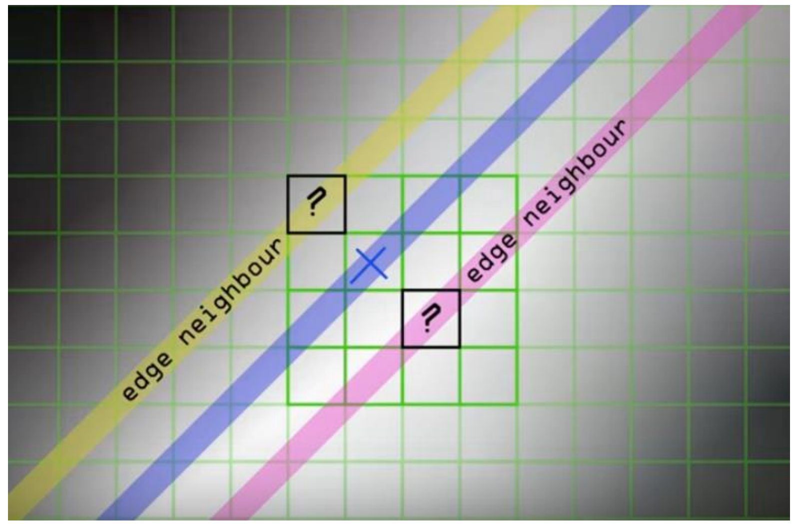
1. find the direction, $\hat{d}_k$, which best approximates the direction $E_o(i, j)$ (the normal to the edge);
2. if $E_s(i, j)$ is smaller than at least one of its two neighbors along $\hat{d}_k$, assign $I_N(i, j) = 0$ (suppression); otherwise assign $I_N(i, j) = E_s(i, j)$.

The output is an image, $I_N(i, j)$, of the thinned edge points (that is, $E_s(i, j)$ after suppressing nonmaxima edge points).

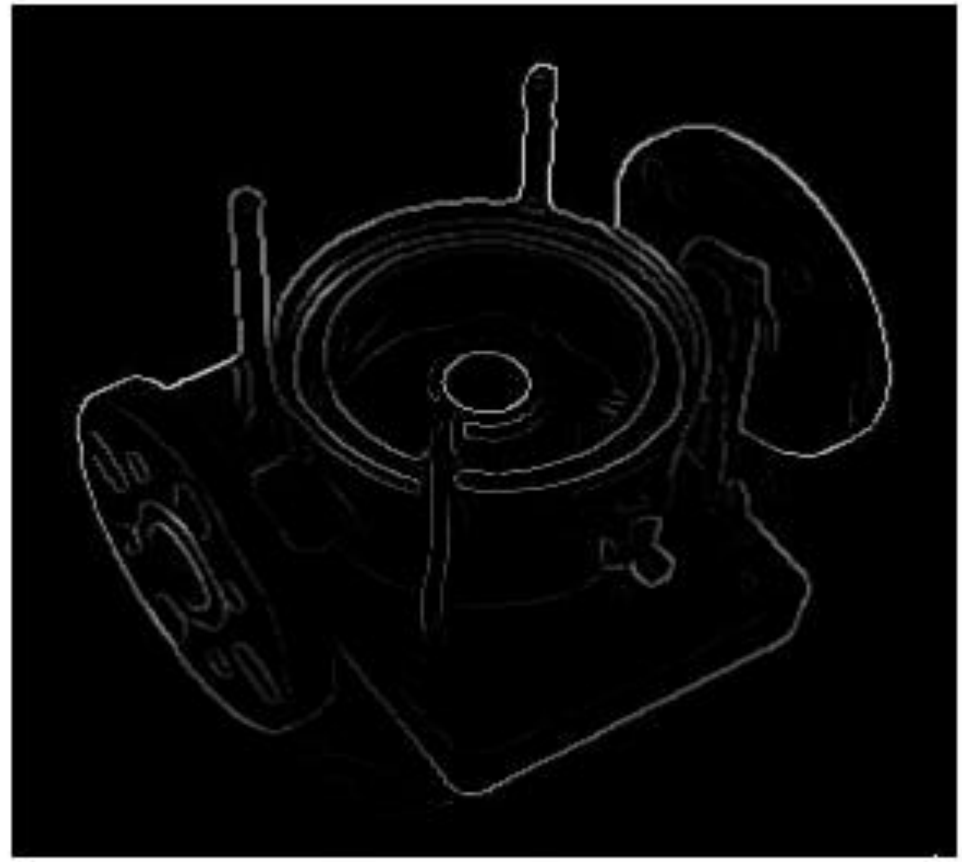
1. Round the gradient direction θ to nearest 45°, corresponding to the use of an 8-connected neighbourhood.
2. Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient direction. I.e. if the gradient direction is north ($theta = 90°$), compare with the pixels to the north and south.
3. If the edge strength of the current pixel is largest; preserve the value of the edge strength. If not, suppress (i.e. remove) the value.




?
?
edge neighbour
edge neighbour


?
?
edge neighbour
edge neighbour

(a) Gradient values



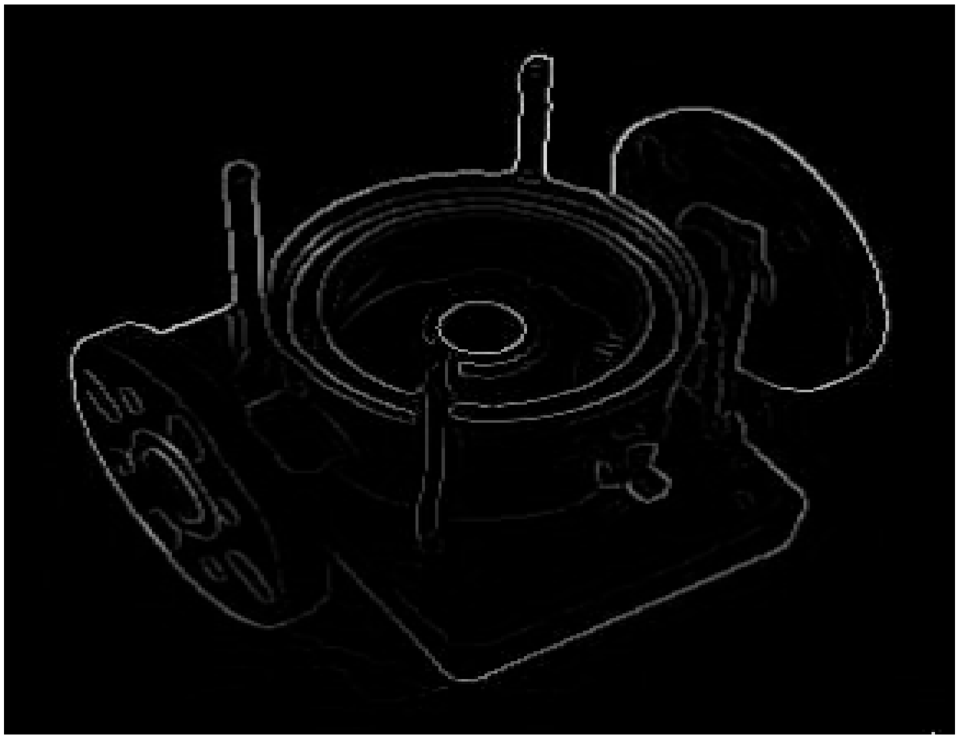
(b) Edges after non-maximum suppression

2.4 Double thresholding

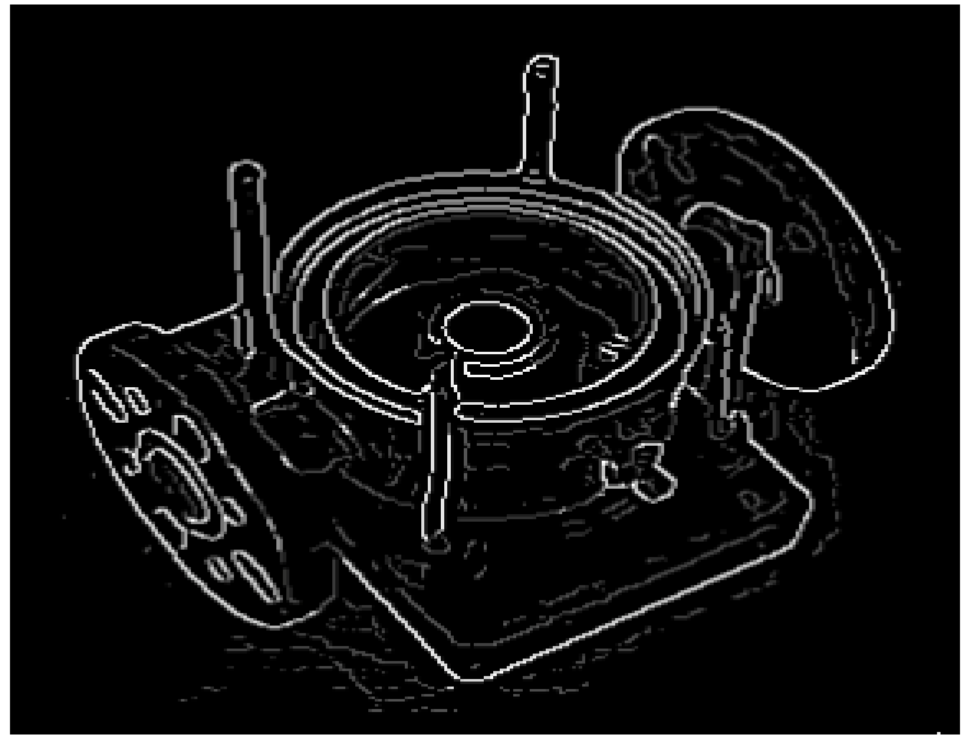
Edge pixels stronger than the high threshold are marked as *strong*; edge pixels weaker than the low threshold are suppressed and edge pixels between the two thresholds are marked as *weak*.

```
} procedure Edge_Detect( Mag[], T_low, T_high, E[] );
{
      for x := 0 to MaxX - 1;
      for y := 0 to MaxY - 1;
         {
           if (Mag[x, y] ≥ T_high) then Follow_Edge( x, y, Mag[], T_low, T_high, E[] );
         } ;
}
```

$\mathbf{I}[\mathbf{x}, \mathbf{y}]$: input intensity image; σ : spread used in Gaussian smoothing;
$\mathbf{E}[\mathbf{x}, \mathbf{y}]$: output binary image;
$\mathbf{IS}[\mathbf{x}, \mathbf{y}]$: smoothed intensity image;
$\mathbf{Mag}[\mathbf{x}, \mathbf{y}]$: gradient magnitude; $\mathbf{Dir}[\mathbf{x}, \mathbf{y}]$: gradient direction;
T_{low} is low intensity threshold; T_{high} is high intensity threshold;



(a) Edges after non-maximum suppression



(b) Double thresholding

2.5 Edge tracking by hysteresis

Strong edges are interpreted as "certain edges", and can immediately be included in the final edge image. Weak edges are included if and only if they are connected to strong edges.

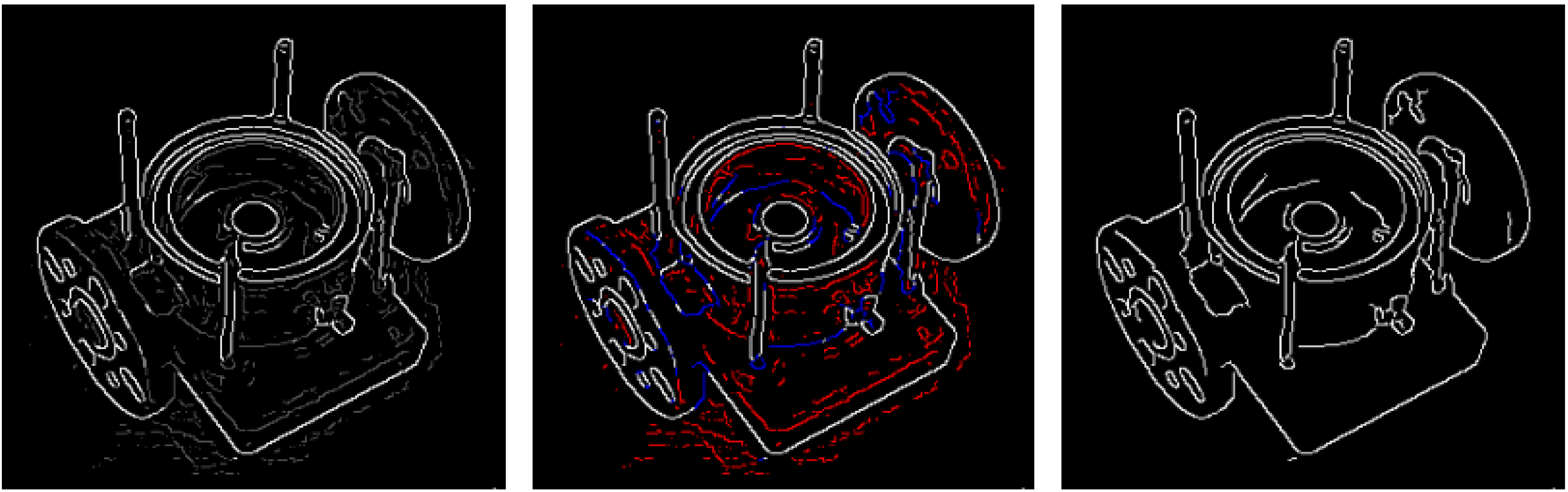
procedure Follow_Edge($x, y, \mathbf{Mag}[], T_{low}, T_{high}, \mathbf{E}[]$);
{
  $\mathbf{E}[x, y] := 1;$
  while $\mathbf{Mag}[u, v] > T_{low}$ for some 8-neighbor $[u, v]$ of $[x, y]$
    {
      $\mathbf{E}[u, v] := 1;$
      $[x, y] := [u, v];$
    } ;
}

$\mathbf{I}[\mathbf{x}, \mathbf{y}]$: input intensity image; σ : spread used in Gaussian smoothing;
$\mathbf{E}[\mathbf{x}, \mathbf{y}]$: output binary image;
$\mathbf{IS}[\mathbf{x}, \mathbf{y}]$: smoothed intensity image;
$\mathbf{Mag}[\mathbf{x}, \mathbf{y}]$: gradient magnitude; $\mathbf{Dir}[\mathbf{x}, \mathbf{y}]$: gradient direction;
T_{low} is low intensity threshold; T_{high} is high intensity threshold;

Algorithm 6.5: Hysteresis to filter output of an edge detector

1. Mark all edges with magnitude greater than t_1 as correct.
2. Scan all pixels with edge magnitude in the range $[t_0, t_1]$.
3. If such a pixel borders another already marked as an edge, then mark it too. 'Bordering' may be defined by 4- or 8-connectivity.
4. Repeat from step 2 until stability.

Milan Sonka et al, Image Processing, Analysis, and Machine Vision



Agenda:

Noise

Filtering

Canny Edge detector

Fourier transform

Fourier Domain Mathematics

Fourier Transform 2D - DFT

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

u = 0, 1, 2, ..., N-1
v = 0, 1, 2, ..., M-1

Inverse Fourier Transform

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

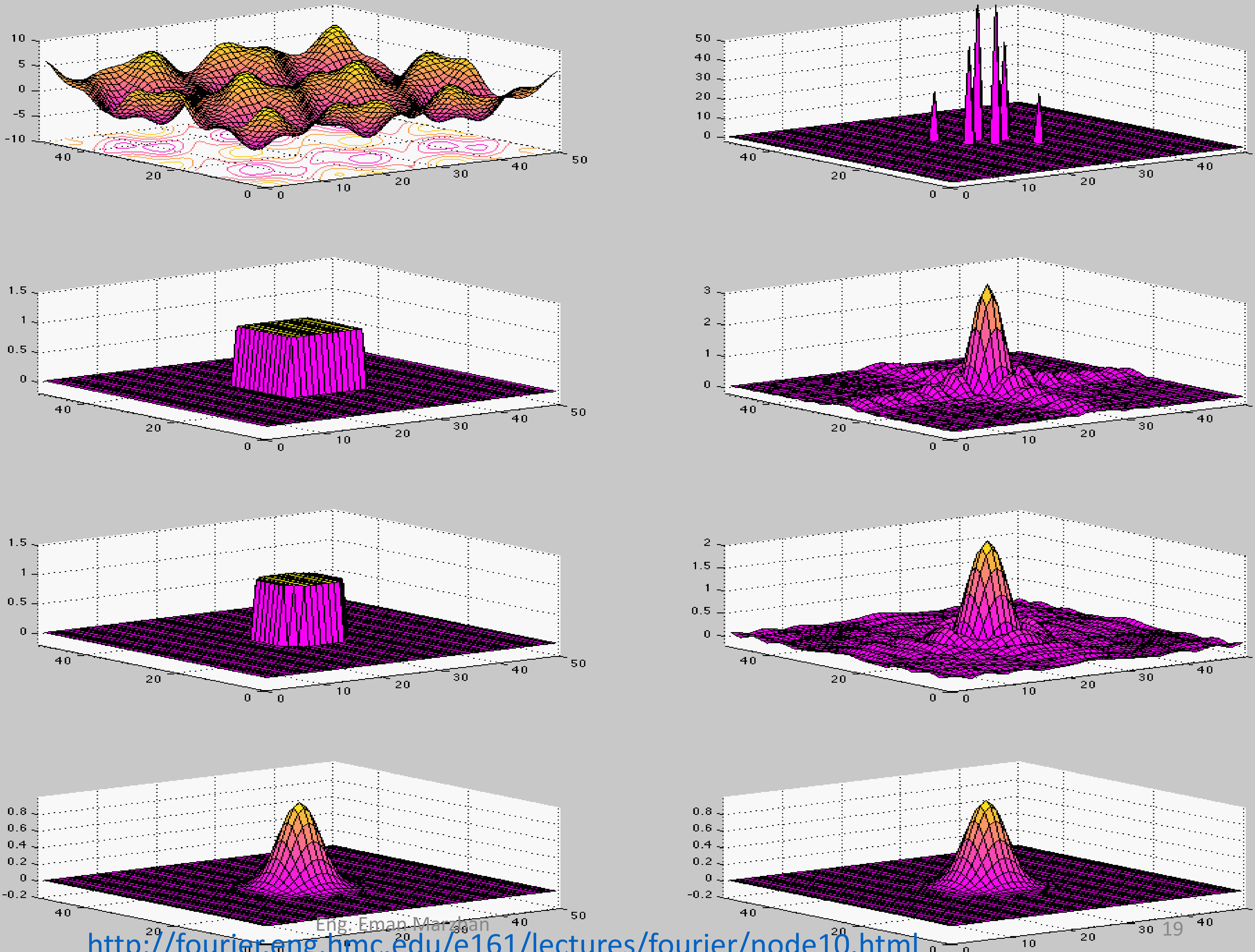
y = 0, 1, 2, ..., N-1
x = 0, 1, 2, ..., M-1

$$\cos x = \mathrm{Re}\{e^{ix}\} = \frac{e^{ix}+e^{-ix}}{2}$$

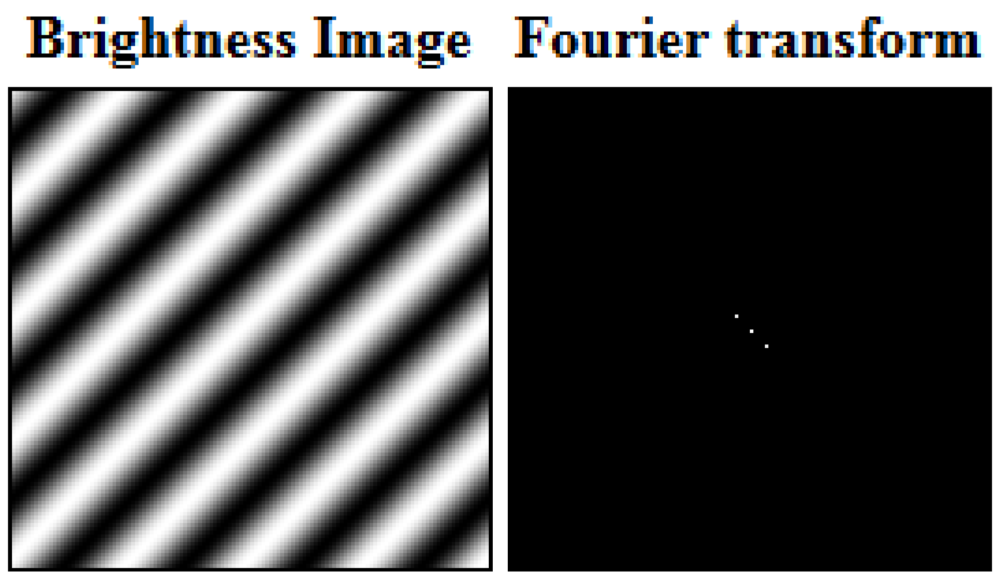
$$\sin x = \mathrm{Im}\{e^{ix}\} = \frac{e^{ix}-e^{-ix}}{2i}$$

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

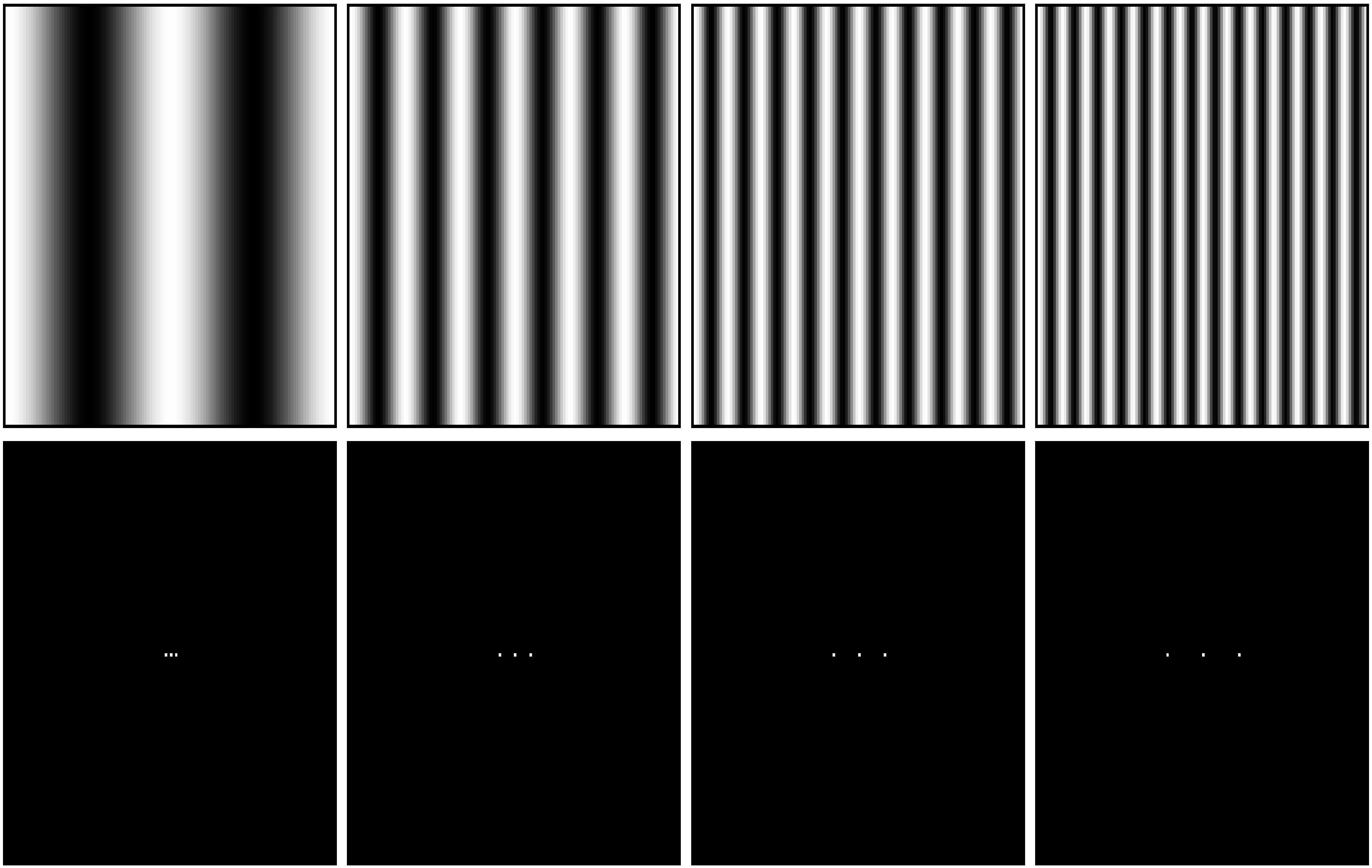
$$e^{-ix} = \cos(-x) + i\sin(-x) = \cos x - i\sin x$$



http://fourier.eng.hmc.edu/e161/lectures/fourier/node10.html


Brightness Image
Fourier transform

Width in space ➔ ?? in freq.



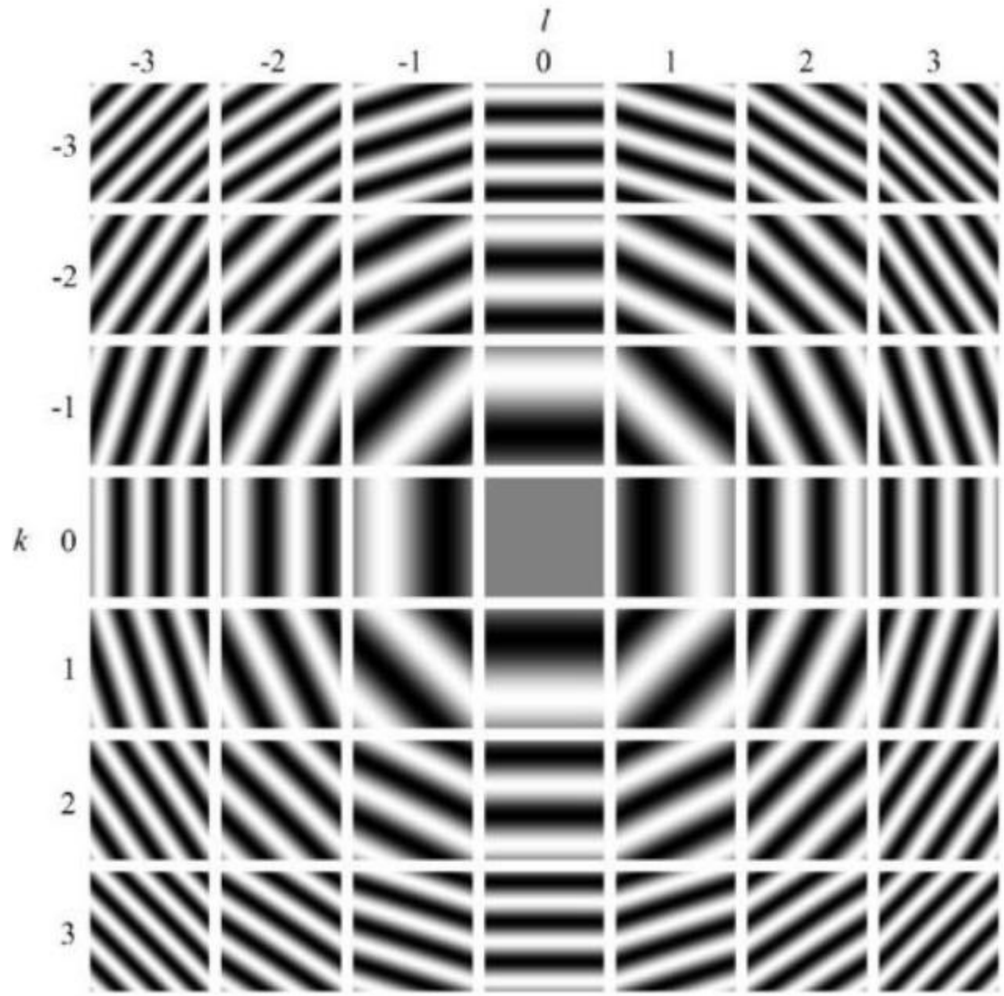
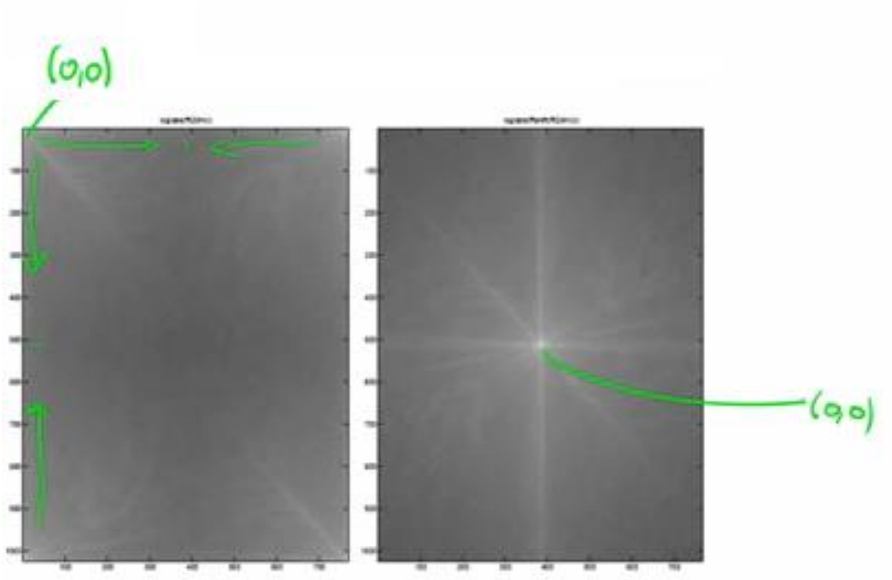


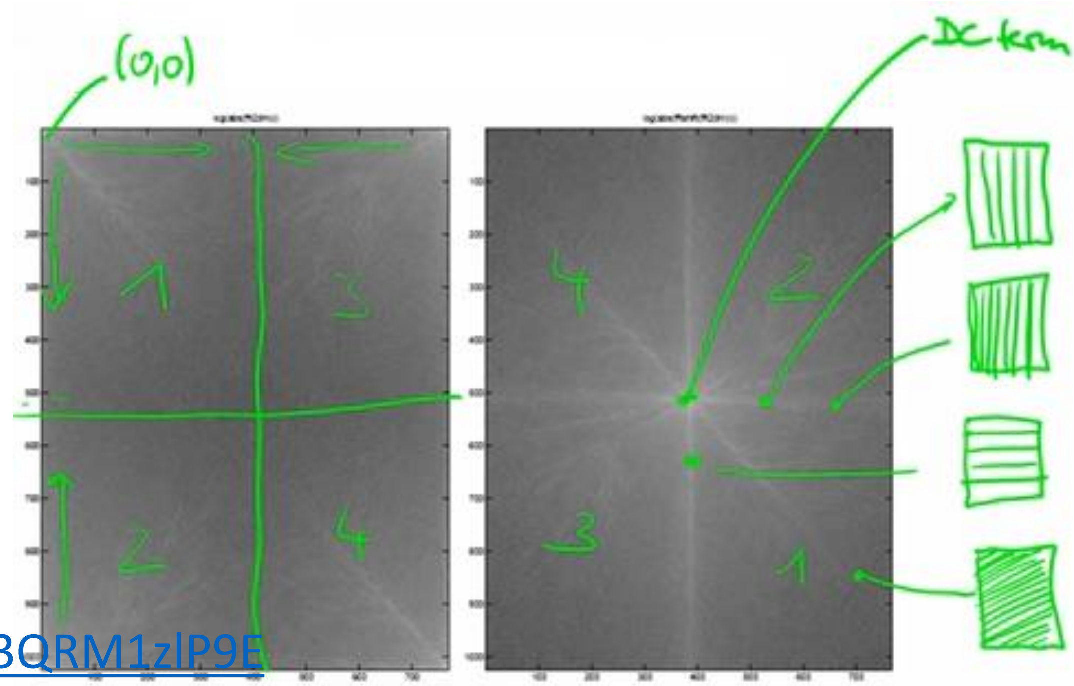
Fig. 4.26 Sine basis functions for the lowest frequencies of a two-dimensional DFT on a squar
matrix. Each panel shows the basis function with vertical frequency *k* and horizontal frequenc
l across the entire spatial image, with greater brightness indicating higher values.

https://www.slideserve.com/jerry-king/the-fourier-transform



Logarithmic

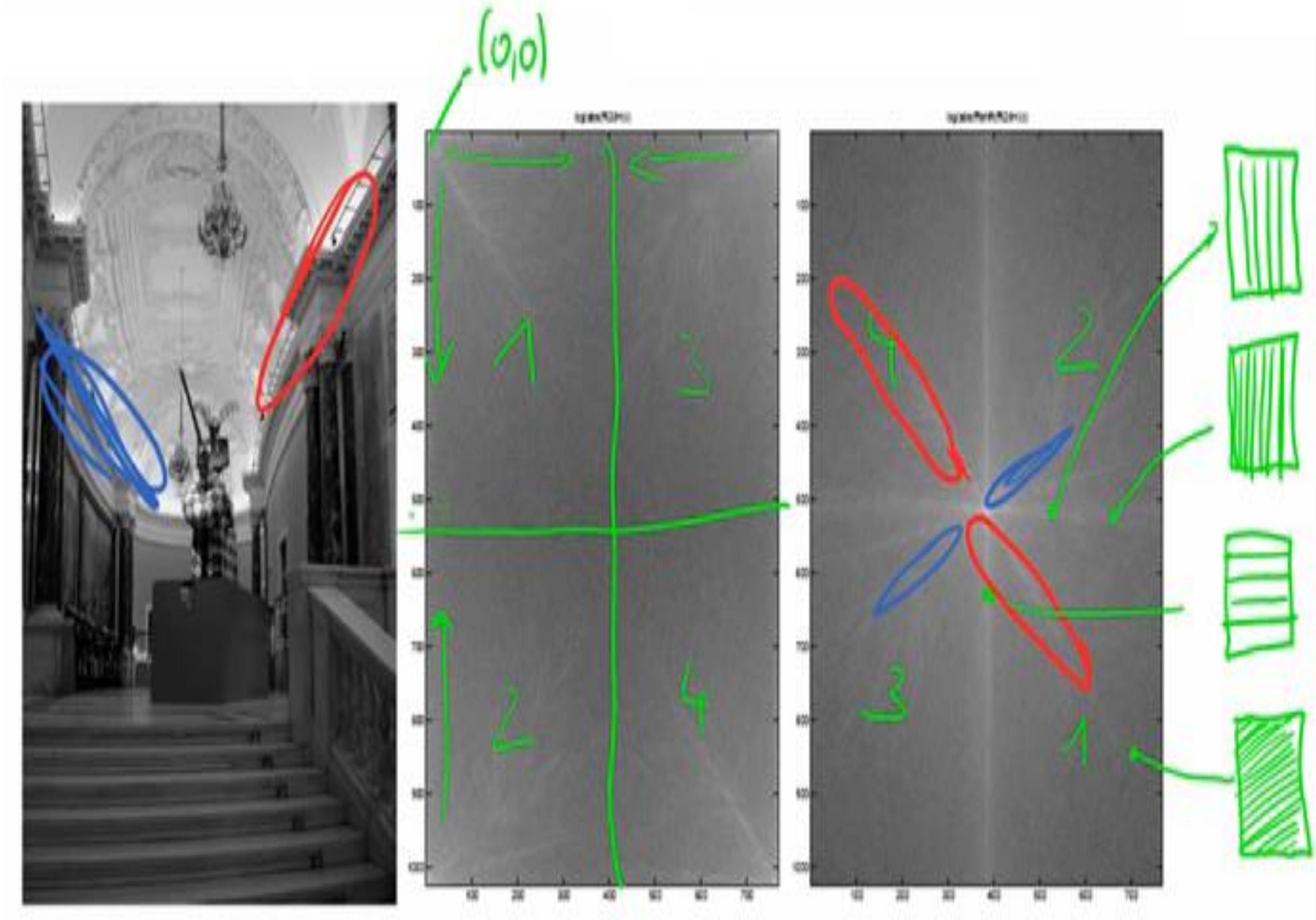
Display Log(mag(F)<u>**+1**</u>) ?



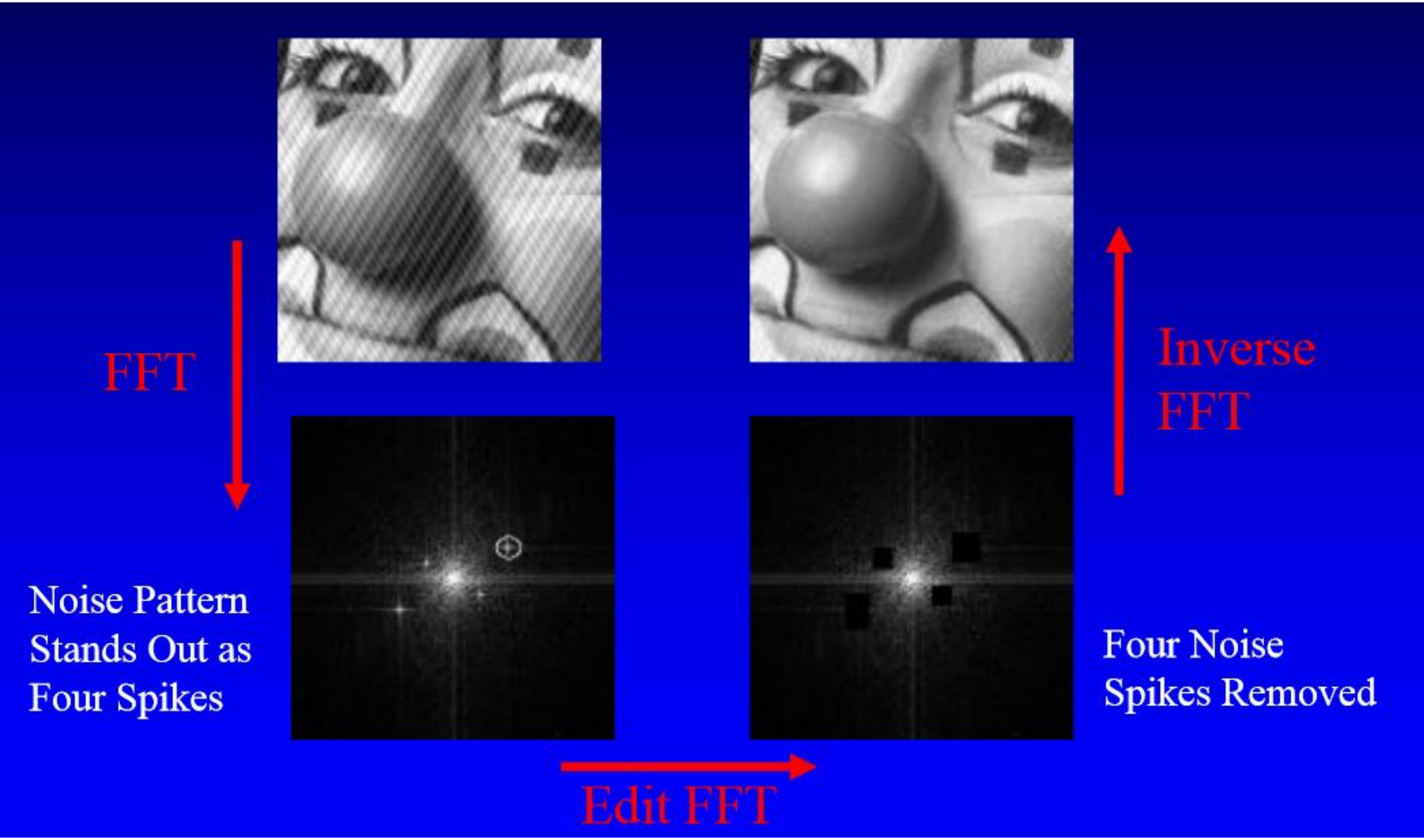
[Universität Heidelberg](https://www.youtube.com/watch?v=-3QRM1zlP9E)

https://www.youtube.com/watch?v=-3QRM1zlP9E

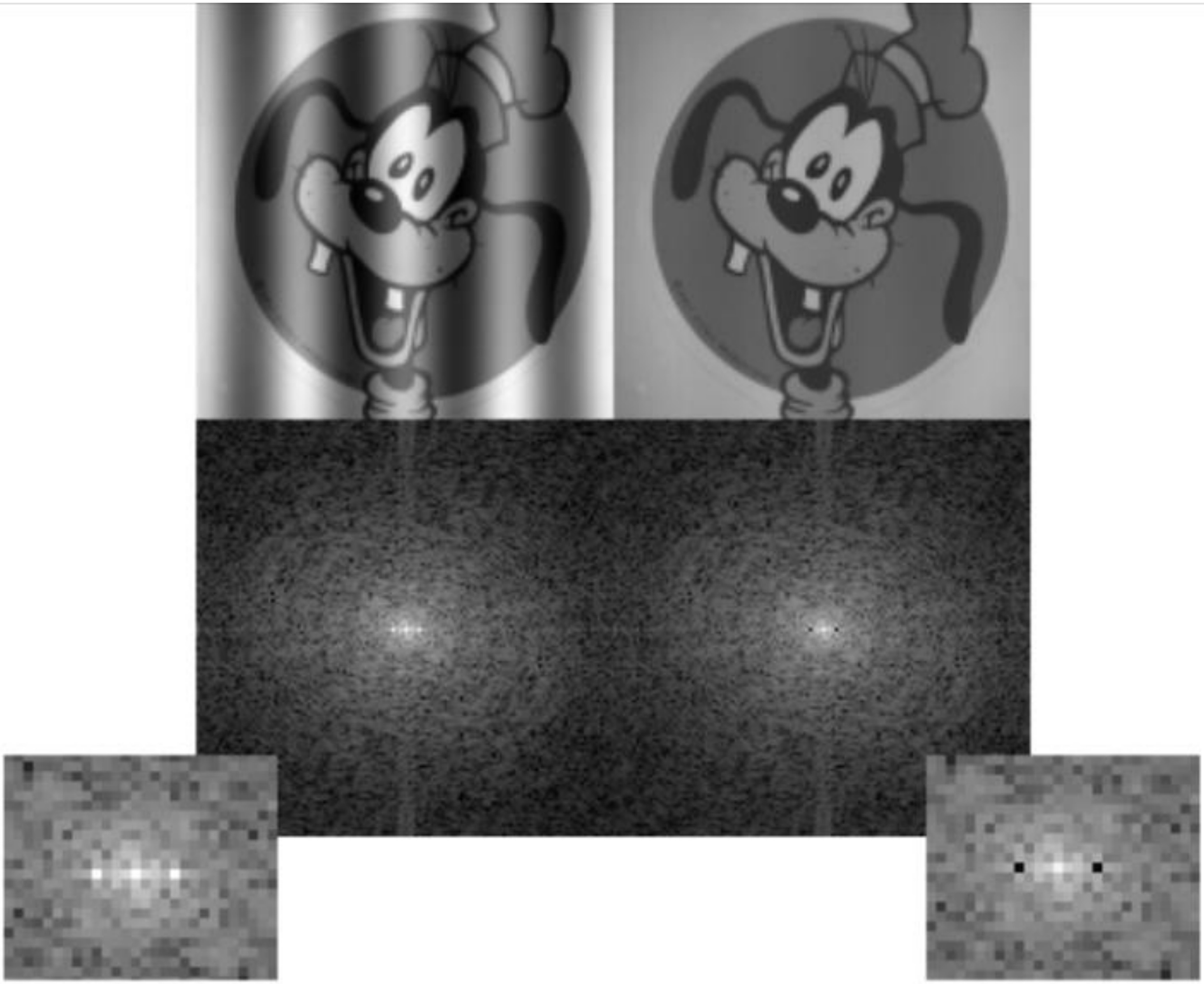
FFT Interpretation



Fourier Domain Applications



Fourier Domain Applications



Fourier Domain Applications

Original image

Power spectrum with mask that filters low frequencies

Result of inverse transform

Power spectrum with mask that passes low frequencies

Result of inverse transform