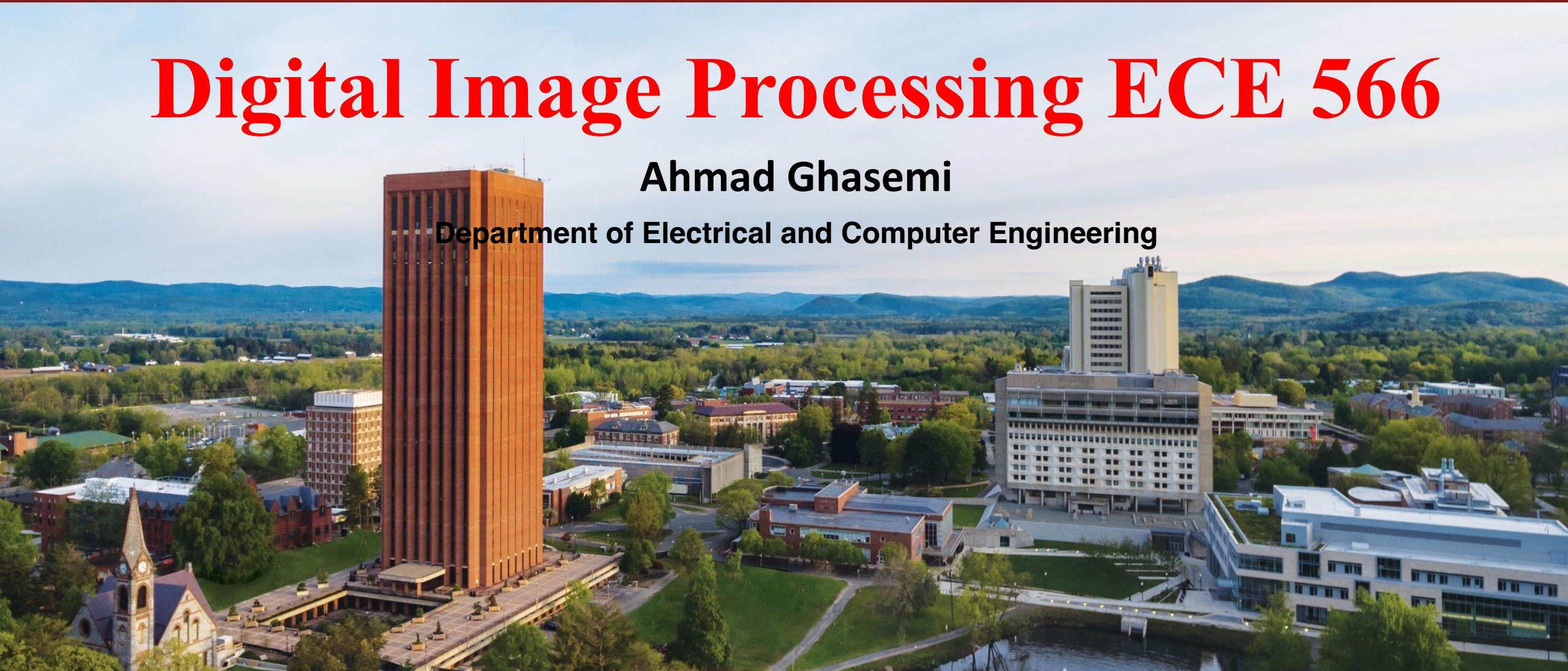
University of Massachusetts Amherst

Digital Image Processing ECE 566

Ahmad Ghasemi

Department of Electrical and Computer Engineering



2D Discrete Fourier Transform

- Or, as follows

$$F[k,l] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m,n] e^{-j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

where $k = 0,1,\ldots,M-1$ and $l = 0,1,\ldots,N-1$

- Inverse DFT

$$f[m,n] = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F[k,l] e^{j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)}$$

2D Discrete Fourier Transform: DC component

$$F(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-j2\pi(\frac{k}{M}m + \frac{l}{N}n)}$$

The value $F(0,0)$ of the DFT is called the dc coefficient

If we put u = v = 0, then

$$F(0,0) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{0} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)$$

Essentially $F(0,0)$ is the sum of all terms in the original matrix/image.

2D Discrete Fourier Transform: Example

Suppose we have as input a constant image of all 1's, $f(m,n) = 1$

The DFT yields just a DC component, 0 everywhere else.

In this example, DC component is sum of all elements = 64

64	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

2D Discrete Fourier Transform: Example

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \ddots & & \vdots \\ 1 & & \cdots & 1 \end{bmatrix}_{8\times 8} = 1 \rightarrow \text{2D DFT} \rightarrow \begin{cases} M = 8 \\ N = 8 \\ f(m,n) = 1 \end{cases}$$

$$\rightarrow F(k,l) = \sum_{m=0}^{7}\sum_{n=0}^{7} e^{-j2\pi\left(\frac{k}{8}m + \frac{l}{8}n\right)}$$

First $F(0,0)$ or DC coefficient $\rightarrow k = l = 0$

$$\rightarrow F(0,0) = \sum_{m=0}^{7}\sum_{n=0}^{7} e^{0} = \sum_{m=0}^{7}\sum_{n=0}^{7} 1 = 64$$

2D Discrete Fourier Transform: Example

Next, $k \neq 0$ & $l \neq 0$

$$\rightarrow F(k,l) = \sum_{m=0}^{7} \sum_{n=0}^{7} e^{-j2\pi\left(\frac{km}{8} + \frac{ln}{8}\right)}$$

$$= \sum_{m=0}^{7} \sum_{n=0}^{7} e^{-j2\pi \times \frac{km}{8}} \cdot e^{-j2\pi \frac{ln}{8}}$$

$$= \underbrace{\sum_{m=0}^{7} e^{-j2\pi \times \frac{km}{8}}}_{A} \cdot \underbrace{\sum_{n=0}^{7} e^{-j2\pi \frac{ln}{8}}}_{B}$$

2D Discrete Fourier Transform: Example

Note: $\sum_{i=0}^{I} e^{\alpha i} = \frac{1-e^{\alpha(I+1)}}{1-e^{\alpha}}$

Thus, $A = \frac{1-e^{-j2\pi \times \frac{K}{8}(8)}}{1-e^{-j2\pi\frac{k}{8}}} = \frac{1-e^{-j2\pi k}}{1-e^{-j\frac{\pi k}{4}}}$

$1-e^{-j2\pi k} = 1-(\cos(2\pi k) - j\sin(2\pi k))$

$= 1-(1-j0) = 0$ (*)

k is an integer

2D Discrete Fourier Transform: Example

$$B = \frac{1 - e^{-j2\pi \times \frac{l}{8} \times 8}}{1 - e^{-j2\pi \frac{l}{8}}} = \frac{1 - e^{-j2\pi l}}{1 - e^{-j\frac{\pi l}{4}}} = 0$$

(**)

Thus $\xrightarrow[(**)]{(*)} F(k,l) = 0 \quad \forall k \neq 0, l \neq 0$

2D Discrete Fourier Transform: Example

$$f(m,n) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad F(k,l) = ?$$

In class example

University of Massachusetts Amherst

2D Discrete Fourier Transform: Python

https://www.unioviedo.es/compnum/labs/PYTHON/lab06_Fourier2D.html#contents

2D Continuous Fourier Transform

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}\,dx\,dy,$$
$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{j2\pi(ux+vy)}\,du\,dv$$

where u and v are spatial frequencies.

Also will write FT pairs as $f(x,y) \Leftrightarrow F(u,v)$.

- $F(u,v)$ is complex in general,

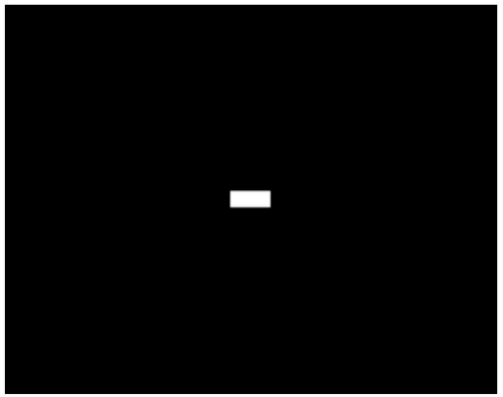
$$F(u,v) = F_{\mathrm{R}}(u,v) + jF_{\mathrm{I}}(u,v)$$

- $|F(u,v)|$ is the magnitude spectrum
- $\arctan(F_{\mathrm{I}}(u,v)/F_{\mathrm{R}}(u,v))$ is the phase angle spectrum.
- Conjugacy: $f^*(x,y) \Leftrightarrow F(-u,-v)$
- Symmetry: $f(x,y)$ is even if $f(x,y) = f(-x,-y)$

University of Massachusetts Amherst

2D Continuous Fourier Transform: Example

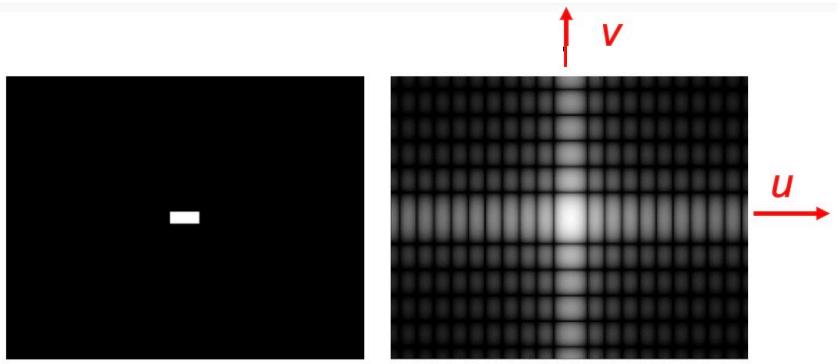
Rectangle centered at origin with sides of length X and Y

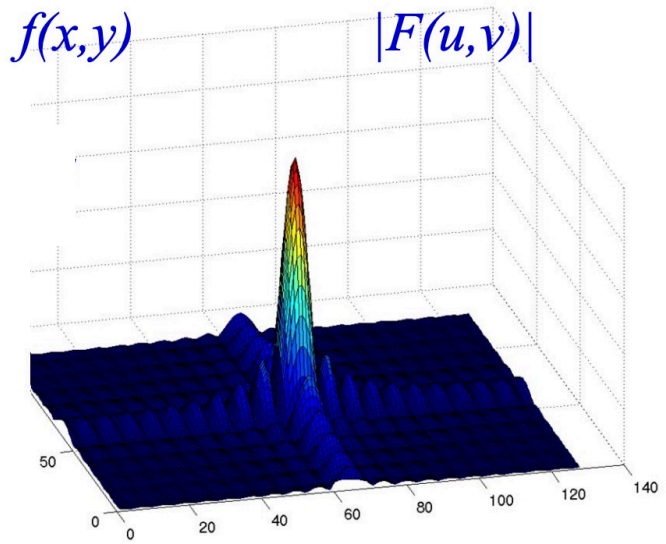


2D Continuous Fourier Transform: Example

Rectangle centered at origin with sides of length X and Y

$$
\begin{aligned}
F(u,v) &= \int\int f(x,y)e^{-j2\pi(ux+vy)}dxdy, \\
&= \int_{-X/2}^{X/2} e^{-j2\pi ux}dx \int_{-Y/2}^{Y/2} e^{-j2\pi vy}dy, \\
&= \left[\frac{e^{-j2\pi ux}}{-j2\pi u}\right]_{-X/2}^{X/2} \left[\frac{e^{-j2\pi vy}}{-j2\pi v}\right]_{-Y/2}^{Y/2}, \\
&= \frac{1}{-j2\pi u}\left[e^{-juX} - e^{juX}\right]\frac{1}{-j2\pi v}\left[e^{-jvY} - e^{jvY}\right], \\
&= XY\left[\frac{sin(\pi Xu)}{\pi Xu}\right]\left[\frac{sin(2\pi Yv)}{\pi Yv}\right] \\
&= XY\mathrm{sinc}(\pi \mathrm{Xu})\mathrm{sinc}(\pi \mathrm{Yv}).
\end{aligned}
$$





|F(u,v)|

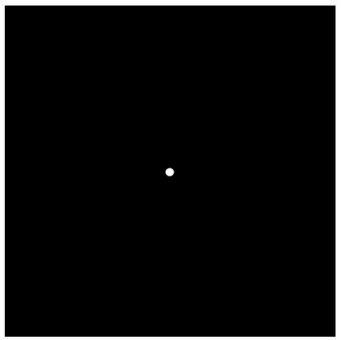
University of Massachusetts Amherst

2D Continuous Fourier Transform: 2 more examples

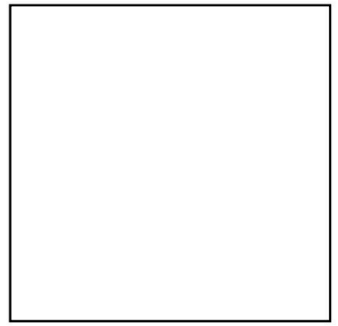
$$f(x,y) = \delta(x,y) = \delta(x)\delta(y)$$

$$F(u,v) = \int\int \delta(x,y) e^{-j2\pi(ux+vy)}\, dxdy$$

$$= 1$$

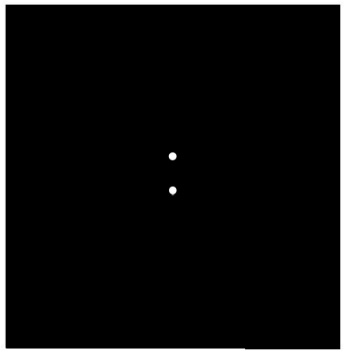


f(x,y)



F(u,v)

$$f(x,y) = \frac{1}{2}(\delta(x,y-a) + \delta(x,y+a))$$



$$F(u,v) = \frac{1}{2}\int\int (\delta(x,y-a) + \delta(x,y+a))\, e^{-j2\pi(ux+vy)}\, dxdy$$

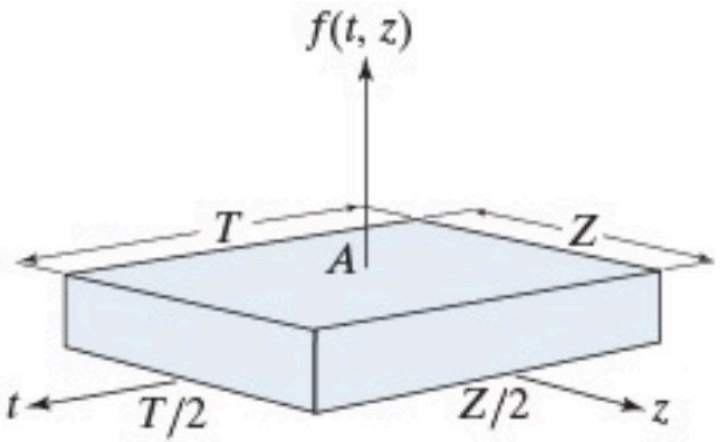
$$= \frac{1}{2}\left(e^{-j2\pi av} + e^{j2\pi av}\right) = \cos 2\pi av$$

University of Massachusetts Amherst

2D Continuous Fourier Transform: Example

$$F(\mu,\nu) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)e^{-j2\pi(\mu t + \nu z)}\,dt\,dz$$

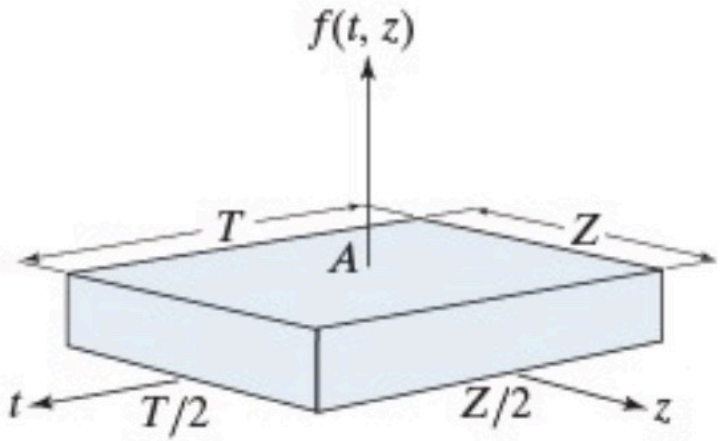
$$f(t,z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\mu,\nu)e^{j2\pi(\mu t + \nu z)}\,d\mu\,d\nu$$

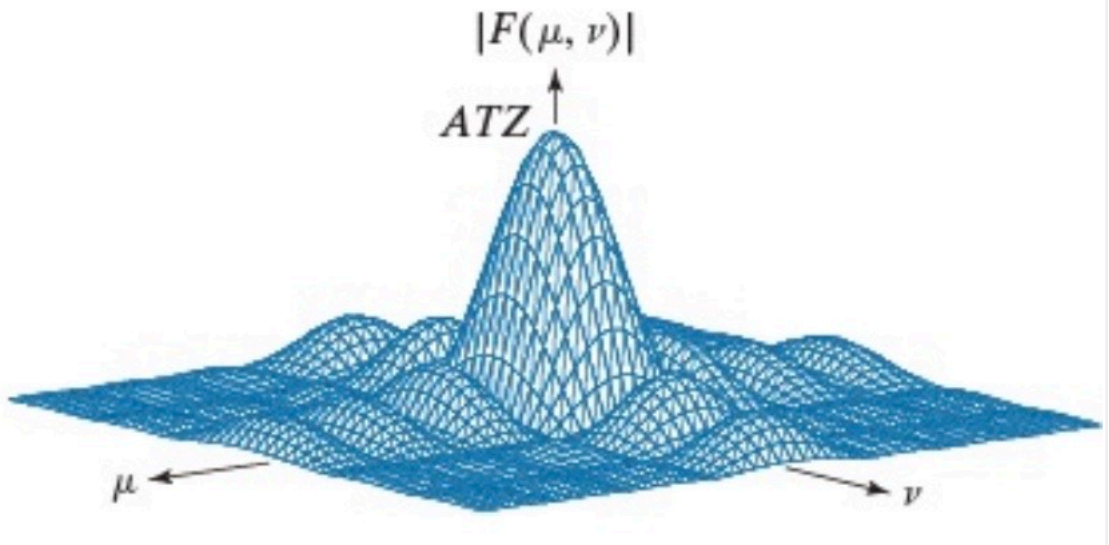


In class example

University of Massachusetts Amherst

2D Continuous Fourier Transform: Example





$$F(\mu,\nu) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)e^{-j2\pi(\mu t+\nu z)}\,dt\,dz = \int_{-T/2}^{T/2}\int_{-Z/2}^{Z/2} Ae^{-j2\pi(\mu t+\nu z)}\,dt\,dz$$

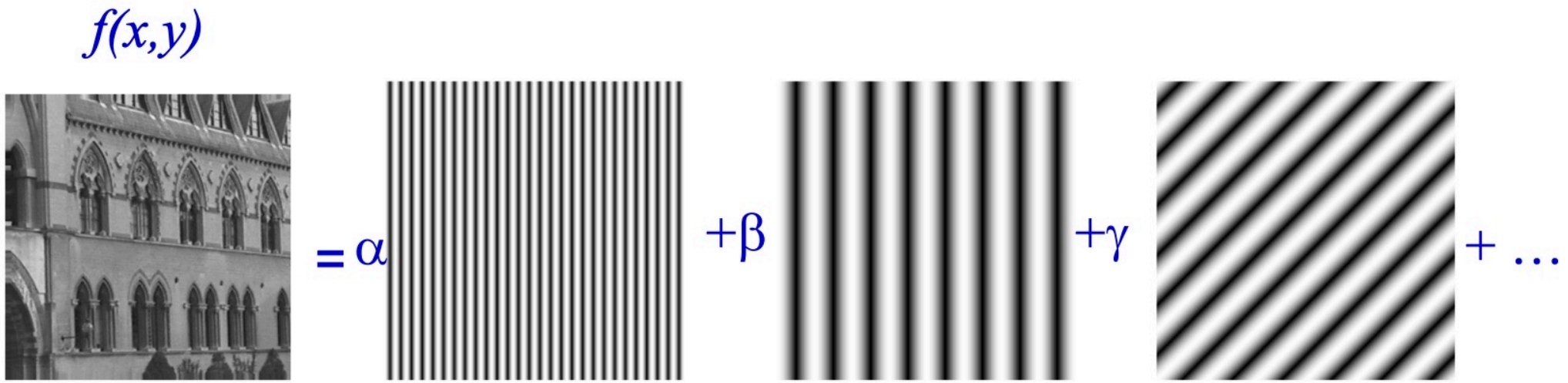
$$= ATZ\left[\frac{\sin(\pi\mu T)}{(\pi\mu T)}\right]\left[\frac{\sin(\pi\nu Z)}{(\pi\nu Z)}\right]$$

2D Continuous Fourier Transform: Summary

The spatial function $f(x, y)$

$$f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u,v) e^{j2\pi(ux+vy)} \, du \, dv$$

is decomposed into a weighted sum of 2D orthogonal basis functions in a similar manner to decomposing a vector onto a basis using scalar products.



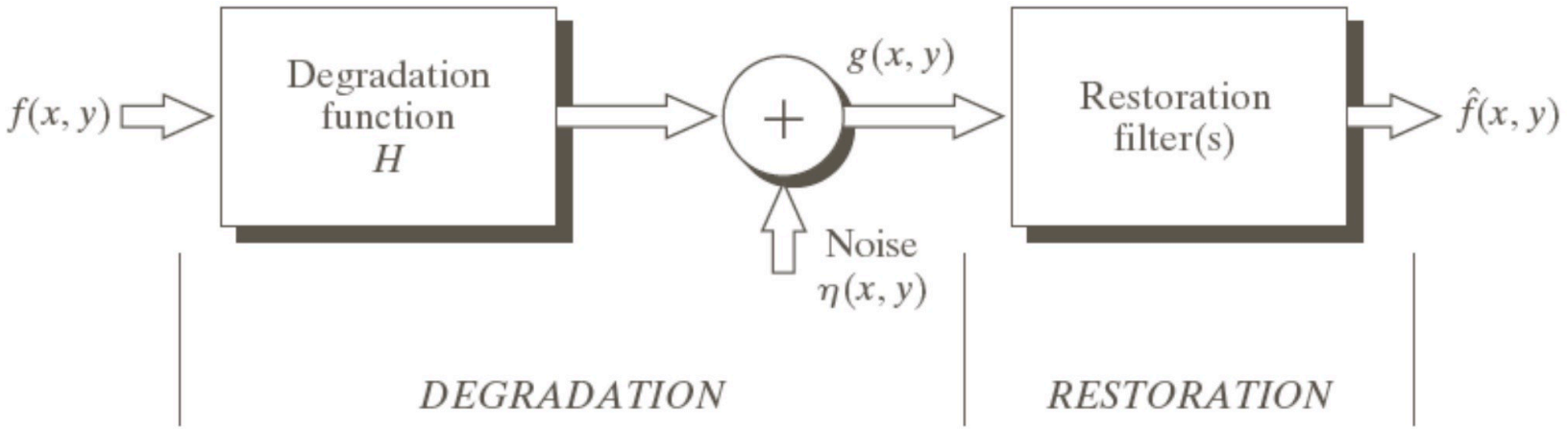
2D Continuous Fourier Transform: Python

https://thepythoncodingbook.com/2021/08/30/2d-fourier-transform-in-python-and-fourier-synthesis-of-images/

Image Restoration

Restoration: reconstruct/recover a degraded image

Model of Image Degradation/Restoration

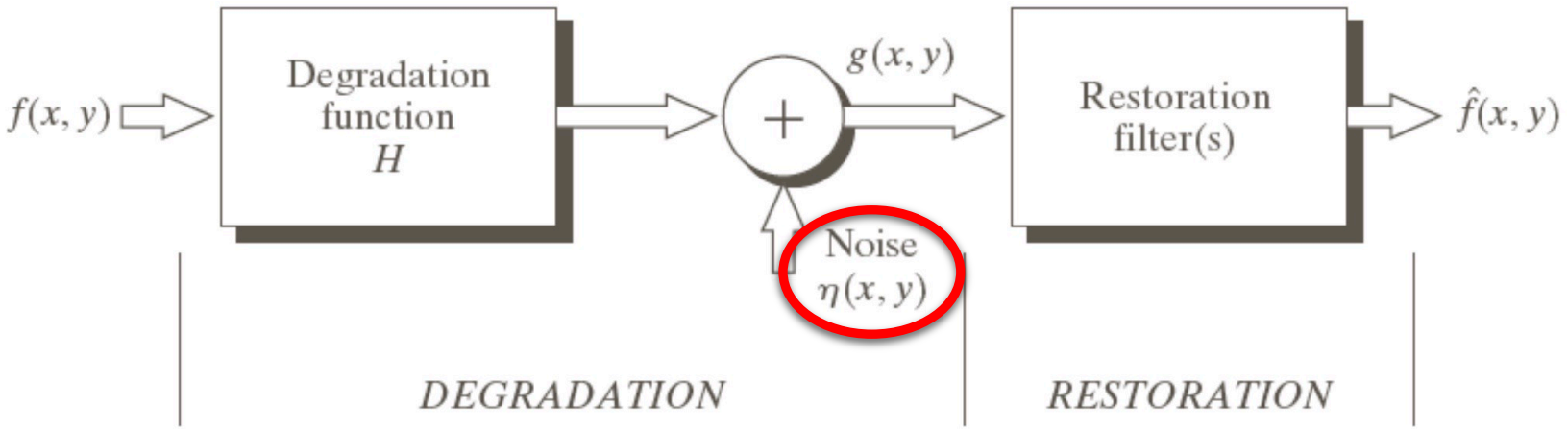


$$g(x,y) = h(x,y) \star f(x,y) + \eta(x,y)$$

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

University of Massachusetts Amherst

Model of Image Degradation/Restoration



$$g(x,y) = h(x,y) \star f(x,y) + \eta(x,y)$$

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

Noise Models

- ✓ Spatial noise models are based on the statistical behavior of gray-level values

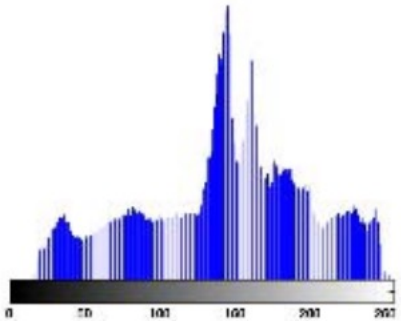
- ✓ Gray-level values are considered as random variables characterized by a probability density function (PDF)

  - ✓ Gaussian (normal)
  - ✓ Impulse (salt-and-pepper)
  - ✓ Uniform
  - ✓ Rayleigh
  - ✓ Gamma (Erlang)
  - ✓ Exponential

Noise Estimation

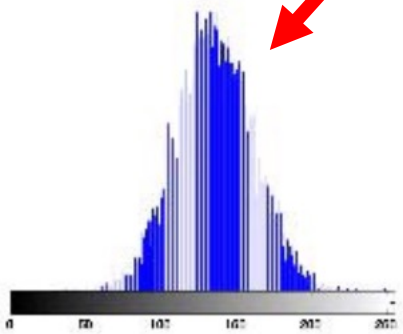
Histogram of

Original image

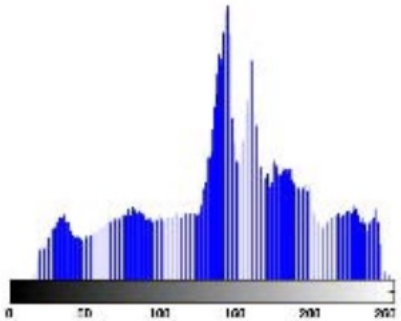


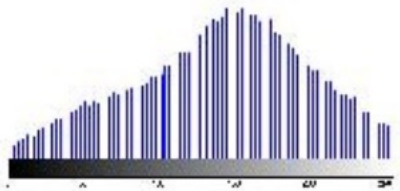
University of Massachusetts Amherst

Noise Estimation



Histogram of Original image





Histogram of Noisy image

Restoration of Noise-only Degradation

Filters to be considered

Mean filters:

- Arithmetic mean filter
- Geometric mean filter
- Harmonic mean filter
- Contraharmonic mean filter

Order-statistics filters:

- Median filter
- Max and min filters
- Midpoint filter
- Alpha-trimmed mean filter

University of Massachusetts Amherst

Mean filters: Arithmetic Mean Filters

- Smooth local variations in an image
- Noise is reduced as a result of blurring
- $g(s,t)$: degraded image
- S_{xy}: set of coordinates in a rectangular subimage window of size $m \times n$

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t)$$

- Causes a certain amount of blurring (proportional to the window size) to the image, thereby reducing the effects of noise.
- Can be used to reduce noise of different types, but works best for Gaussian, uniform, or Erlang noise.

University of Massachusetts Amherst

Mean filters: Arithmetic Mean Filters (Example)

$$\begin{bmatrix} 1 & 2 & \cdots & 5 \\ \vdots & & \ddots & \vdots \\ 1 & 2 & \cdots & 5 \end{bmatrix}_{5\times 5}$$

Mean filters: Geometric Mean Filters

- Comparable to the arithmetic mean filter→ achieve smoothing; lose less image detail in the process

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

- A variation of the arithmetic mean filter
- Primarily used on images with Gaussian noise
- Retains image detail better than the arithmetic mean

Mean filters: Geometric Mean Filters (Example)

$$\begin{bmatrix} 1 & 2 & \cdots & 5 \\ \vdots & & \ddots & \vdots \\ 1 & 2 & \cdots & 5 \end{bmatrix}_{5\times 5}$$

Mean filters: Harmonic Mean Filters

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

- Another variation of the arithmetic mean filter
- Useful for images with Gaussian or salt noise
- Black pixels (pepper noise) are not filtered

Mean filters: Harmonic Mean Filters (Example)

$$\begin{bmatrix} 1 & 2 & \cdots & 5 \\ \vdots & & \ddots & \vdots \\ 1 & 2 & \cdots & 5 \end{bmatrix}_{5\times 5}$$

Mean filters: Contra-harmonic Mean Filters

- Reduce the effects of salt-and-pepper noise
- Q>0→ eliminate pepper noise (Q: order of the filter)
- Q<0→ eliminate salt noise
- Q=0→ arithmetic mean filter
- Q=−1→ harmonic mean filter
- Cannot eliminate both the salt noise and pepper noise simultaneously

$$\hat{f}(x,y)=\frac{\sum_{(s,t)\in S_{xy}} g^{Q+1}(s,t)}{\sum_{(s,t)\in S_{xy}} g^{Q}(s,t)}$$

University of Massachusetts Amherst

Mean filters: Contra-harmonic Mean Filters (Example)

$$\begin{bmatrix} 1 & 2 & \cdots & 5 \\ \vdots & & \ddots & \vdots \\ 1 & 2 & \cdots & 5 \end{bmatrix}_{5\times 5}$$

Mean filters features

- The positive-order filter → effectively reduce the pepper noise, at the expense of blurring the dark areas
- The negative-order filter → effectively reduce the salt noise, at the expense of blurring the bright areas
- Arithmetic and geometric mean filters → suit the Gaussian or uniform noise (particularly the geometric mean filter)
- Contraharmonic filter → suit the impulse noise, yet, with the information of dark or light noise to select the proper sign for Q