# 2D Fourier Transform

Week IV

2-D DFT & Properties



## Fourier Transform - review

1-D:

$$F(u) \equiv \Im\{f(x)\} = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

$$f(x) \equiv \Im^{-1}\{F(u)\} = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

2-D:

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

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



## 2D FT: Properties

Linearity: a f(x,y) + b g(x,y) $\Leftrightarrow$ a F(u,v) + b G(u,v)

Convolution: f(x,y) $\star$ g(x,y) = F(u,v) G(u,v)

Multiplication: f(x,y) g(x,y) = F(u,v) $\star$G(u,v)

Separable functions: Suppose f(x,y) = g(x) h(y), Then F(u,v)=G(u)H(v)

Shifting: $f(x \pm x_0, y \pm y_0)$ $\leftrightarrow$ $\exp[2\pi (x_0 u \pm y_0 v)]$ F(u,v)



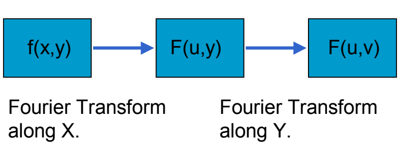
## Separability of the FT

$$F(u,v) = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(x,y) e^{-j2\pi ux} dx\right] e^{-j2\pi vy} dy$$

$$= \int_{-\infty}^{\infty} F(u,y) \, e^{-j2\pi vy} dy$$



## Separability (contd.)




**<u>We can implement the 2D Fourier transform as a sequence of 1-D Fourier transform operations.</u>**



## Eigenfunctions of LSI Systems

A function f(x,y) is an Eigenfunction of a system T if T[ f(x,y) ] = $\alpha$ f(x,y) for some constant (Possibly complex) $\alpha$.

**For LSI systems, complex exponentials of the form exp{ j2$\pi$ (ux+vy) }, for any (u,v), are the Eigenfunctions.**

## Impulse Response and Eigenfunctions

Consider a LSI system with impulse response h(x,y).
Its output to the complex exponential is

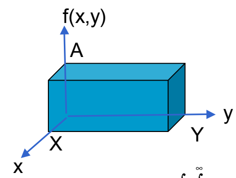
$$g(x,y)=\int\int_{-\infty}^{\infty} h(x-s,y-t)e^{j2\pi(us+vt)}ds\,dt$$

$$=\iint h(\bar{x},\bar{y})e^{j2\pi(ux+vy)}e^{-j2\pi(u\bar{x}+v\bar{y})}d\bar{x}\,d\bar{y}$$

$$=H(u,v)e^{j2\pi(ux+vy)}$$



## 2-D FT: Example



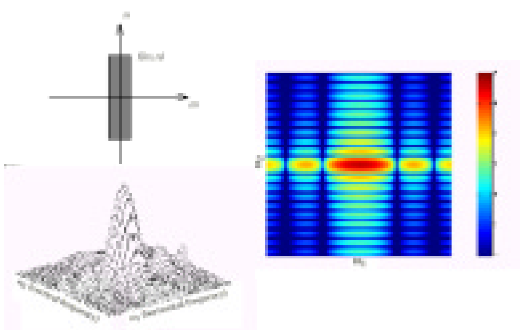

$$F(u,v)=\int\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}dx\,dy$$

$$=A\int_0^X e^{-j2\pi ux}dx\int_0^Y e^{-j2\pi vy}dy$$

$$=AXY\left[\frac{\sin\pi uX}{\pi uX}\right]\left[\frac{\sin\pi vY}{\pi vY}\right]e^{-j\pi(uX+vY)}$$

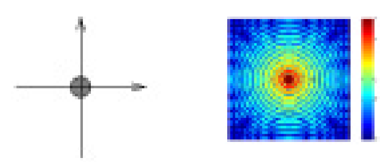


## Example (contd.)





## Example2





## Discrete Fourier Transform

**Consider a sequence {u(n), n=0,1,2,....., N-1}. The DFT of u(n) is**

$$v(k)=\sum_{n=0}^{N-1}u(n)\,W_N^{\;kn}\,,\qquad k=0,1,\ldots\ldots,N-1$$

Where $W_N=e^{-j\frac{2\pi}{N}}$, and the inverse is given by

$$u(n)=\frac{1}{N}\sum_{k=0}^{N-1}v(k)\,W_N^{\;-kn},\quad n=0,1,\ldots,N-1$$



## 2-D DFT

Often it is convenient to consider a symmetric transform:

$$v(k)=\frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}u(n)W_N^{\;kn}\quad \text{and}$$

$$u(n)=\frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}v(k)W_N^{\;-kn}$$

In 2-D: consider a NXN image

$$v(k,l)=\frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}u(m,n)\,W_N^{\;km}W_N^{\;ln},$$

$$u(m,n)=\frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}v(k,l)\,W_N^{\;-km-ln}$$

## 2D DFT -- PROPERTIES

- Separability
- Translation
- Scaling
- Periodicity and Conjugate Symmetry
- Rotation
- convolution



## Separability

$$v(k,l) = \frac{1}{N}\sum_{m=0}^{N-1} W_N^{km} \sum_{n=0}^{N-1} u(m,n)\, W_N^{ln}$$
$$= \frac{1}{\sqrt{N}}\sum_{m=0}^{N-1} v(m,l)\, W_N^{km}$$

For each ‘m’, v(m,l) is the 1-D DFT with frequency values
l = 0,1,....., N-1



## Separability

**The DFT of a 2-D array can be obtained by first taking the 1-D DFT of each row (or column) and then taking the 1-D DFT of each column (or row).**

**<u>It does not matter if the order of operation is reversed.</u>**

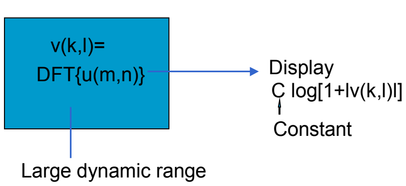


## Translation

$$u(m-m', n-n') \leftrightarrow v(k,l)e^{-j2\pi\frac{(km'+ln')}{N}}$$



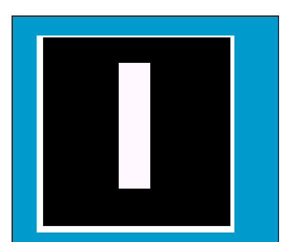
## Displaying the DFT: Scaling





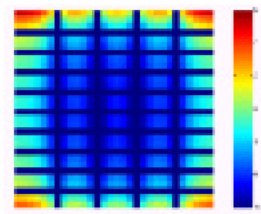

## In MATLAB

**f = zeros(30,30);**
**f(5:24,13:17)=1;**
**imshow(f, ‘notruesize’);**



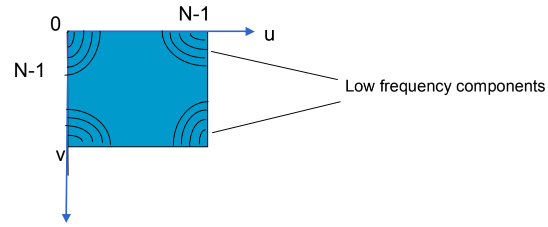
## In Matlab(2)

```
F =fft2(f);
F2 = log(abs(F));
imshow(F2, [-1, 5], 'notruesize');
colormap(jet); colorbar;
```





## Displaying the DFT






## Displaying (again) & Shifting

$$u(m,n)e^{\frac{j2\pi(k'm+l'n)}{N}} \leftrightarrow v(k-k',l-l') \text{ and}$$

$$u(m,n)(-1)^{m+n} \leftrightarrow v\left(k-\frac{N}{2}, l-\frac{N}{2}\right)$$

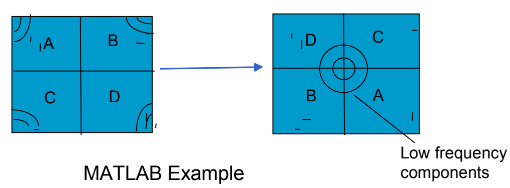
The origin of the F{u(m,n)} can be moved to the center of the array (N X N square) by first multiplying u(m,n) by $(-1)^{m+n}$ and then taking the Fourier transform.

**Note: Shifting does not affect the magnitude of the Fourier transform.**



## Displaying DFT

$|v(k,l)\, e^{-j2\pi[km'+ln']/N}| = |v(k,l)|$

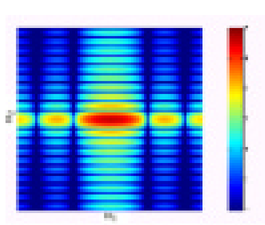



MATLAB Example



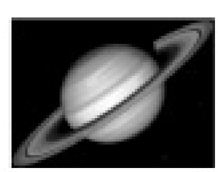
## In Matlab(3): FFTSHIFT

```
F2= fftshift(F);
imshow (log(abs(F2),..)
```

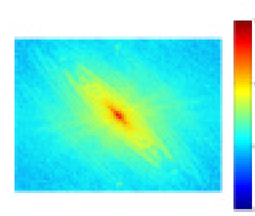




## Another example



Original image



Its centered DFT magnitude

## Periodicity & Conjugate Symmetry

$u(m,n) \overset{F}{\longleftrightarrow} v(k,l)$

$v(k,l) = v(k+N, l) = v(k, l+N) = v(k+N, l+N)$

If u(m,n) is real, v(k,l) also exhibits conjugate symmetry
$v(k,l) = v^*(-k, -l)$ or $|v(k,l)| = |v(-k, -l)|$



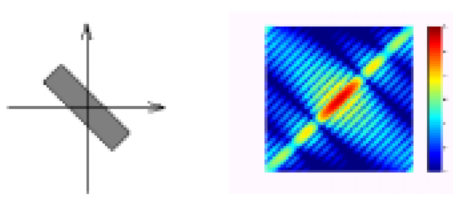
## Rotation

(continuous case)

If you rotate the image u(m,n) by an angle $\theta$, its F.T also gets rotated by the same angle.



## Rotation





## Average Value

$$\bar{u} = \frac{1}{N}\sum_m \sum_n u(m,n) = \text{Average}$$

$$v(k,l) = \frac{1}{N}\sum_m \sum_n u(m,n) e^{-j2\pi \frac{km+ln}{N}}$$

$$v(0,0) = \frac{1}{N}\sum_m \sum_n u(m,n) = N\bar{u}$$

$$\text{or } \bar{u} = \frac{v(0,0)}{N} \quad \text{(Scaled Average)}$$



## Convolution (Revisited)

Consider 1-D continuous case

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(x')\, g(x - x')\, dx'$$

Let $f(x) \leftrightarrow F(u),\ g(x) \leftrightarrow G(u)$

Then $f(x) * g(x) \leftrightarrow F(u)G(u)$

Convolution in Space ⟷ Multiplication in Frequency



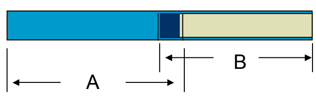
# Discrete Convolution

Let us now assume that we discretize f(x) and g(x) into vectors f(n) and g(n) of lengths A and B

$f(n) \longrightarrow \{f(0), f(1), \ldots.. f(A-1)\}$
$g(n) \longrightarrow \{g(0), g(1), g(2), \ldots. g(B-1)\}$

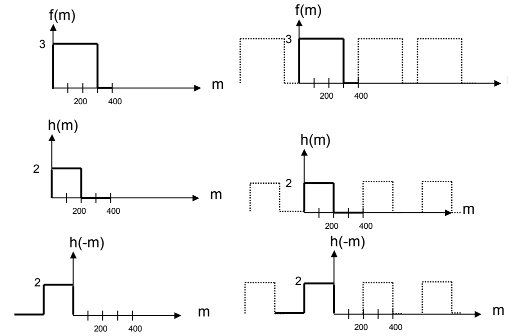
(a) DFT and its inverse are periodic functions
(b) Convolving two vectors of length A and B gives a vector of dimension A+B-1. (Linear convolution)

## Length of the Convolution



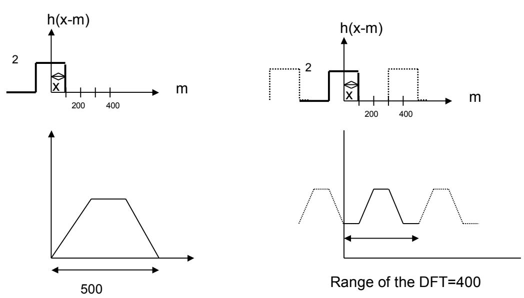



## Discrete Convolution: an example (Fig 4.36)






## Discrete conv. (cont.)






## Zero Imbedding

In order to obtain a convolution theorem for the discrete case, and still be consistent with the periodicity property we need to assume that sequences f( ) and g( ) are periodic with some period M. From (b) it is clear that M>A+B-1 to avoid overlap.

Since this period is greater than A or B, the original sequence length must be increased and this is done by appending zeros at the end. Redefine the extended sequences as

$$f_e(n) = \begin{cases} f(n) & 0 \le n \le A-1 \\ 0 & A \le n \le M-1 \end{cases}$$

$$g_e(n) = \begin{cases} g(n) & 0 \le n \le B-1 \\ 0 & B \le n \le M-1 \end{cases}$$



$$f_e(n) *_c g_e(n) = \sum_{m=0}^{M-1} f_e(m) g_e(n-m)_c$$

where $(g(n))_c = g[n \text{ Modulo } M]$

Note: With n expressed as

$n = n_1 + n_2 N$ where $0 \le n_1 \le N-1$

$n$ modulo $N$ equals $n_1$

$x \mod y = x - y\left[x/y\right]$ if $y \neq 0$

$x \mod 0 = x.$

$\left[\frac{x}{y}\right]$ is the integer part of $x/y$



## Theorem

**The DFT of the circular convolution of two sequences of length N is equal to the product of their DFTs.**

$$\text{If } y(n) = \sum_{m=0}^{N-1} f(n-m)_c g(n) \quad \text{then}$$

$$\text{DFT}[y(n)]_N = \text{DFT}[f(n)]_N \, \text{DFT}[g(n)]_N$$

***A linear convolution of two sequences can be obtained via FFT by embedding it into a circular convolution.***

## 2-D Convolution

**These results can be similarly extended to 2-D signals.**

Let f(m,n) : A x B array
g(m,n) : C x D array

Let M> = A + C -1
N> = B + D -1

For linear convolution using DFT create the extended periodic sequences of period MxN in the 2-D.



## Extended (periodic) Sequences

**computing convolution is more efficient in the frequency domain.**

$$f_e(m,n) = \begin{cases} f(m,n) & 0 \le m \le A-1 \\ & 0 \le n \le B-1 \\ 0 & A \le m \le M-1 \\ & B \le n \le N-1 \end{cases}$$

$$g_e(m,n) = \begin{cases} g(m,n) & 0 \le m \le C-1 \\ & 0 \le n \le D-1 \\ 0 & C \le m \le M-1 \\ & D \le n \le N-1 \end{cases}$$

and the 2 - D linear c onvolution becomes

$$y(m,n) = \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} f_e(m-m', n-n')_c \; g_e(m',n')$$



## Linear Convolution and DFT: Summary

**y(n) = f(n) * g(n)**

1. Let M>= A+B-1 be an integer for which the FFT algorithm is available.

2. Define the zero extended sequences $f_e(n)$, $g_e(n)$.

3. Let $F_e(k)$ = DFT { $f_e(n)$ }$_M$, $G_e(k)$ = DFT { $g_e(n)$ }$_M$. Let $Y_e(k)=F_e(k)G_e(k)$

4. Take the I-DFT of $Y_e(k)$ to obtain $Y_e(n)$.
Then Y (n) = $Y_e(n)$ for 0 <= n <= A+B-1



## A note on convolution with images

**Note: In many cases involving images, we deal with square arrays of size N X N. We normally would like to have the resulting convolved output also as an N X N array.**



## Conv2 (.) in Matlab

CONV2 Two dimensional convolution.

C = CONV2(A, B) performs the 2-D convolution of matrices A and B. If [ma,na] = size(A) and [mb,nb] = size(B), then size(C) = [ma+mb-1,na+nb-1].

C = CONV2( ... ,'shape') returns a subsection of the 2-D convolution with size specified by 'shape':

- 'full' - (default) returns the full 2-D convolution,
- 'same' - returns the central part of the convolution that is the same size as A.
- 'valid' - returns only those parts of the convolution that are computed without the zero-padded edges, size(C) = [ma-mb+1,na-nb+1] when size(A) > size(B).