The Fast Fourier Transform

C306
Fall 2001
Martin Jagersand

DFT in matrix form

$$\begin{bmatrix} F(0) \\ F(2) \\ \vdots \\ F(N-1) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-\frac{j2\pi}{N}} & \cdots & e^{-\frac{j2\pi(N-1)}{N}} \\ \vdots & \ddots & & \vdots \\ 1 & e^{-\frac{j2\pi(N-1)}{N}} & \cdots & e^{-\frac{j2\pi(N-1)^2}{N}} \end{bmatrix} \begin{bmatrix} f(0) \\ f(2) \\ \vdots \\ f(N-1) \end{bmatrix}$$

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-\frac{j2\pi ux}{N}}$$

- Note: u=Ax form!

Inverse:

- y

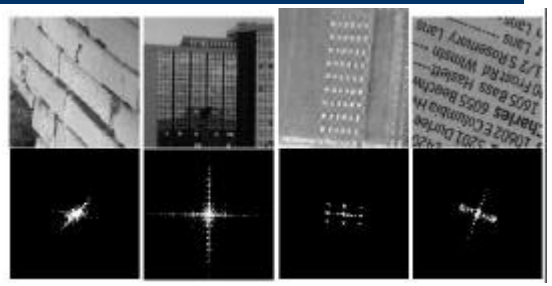
$$A^{T}$$

Displaying the Fourier Transform

- Practically the Fourier transform coefficients have a large range
- Usually the magnitude |F(u,v)| is what we want to see
- Compress range variation to see as an image:

  D(i,j) = c*log(1+|F(u,v)|)

Manmade object images:

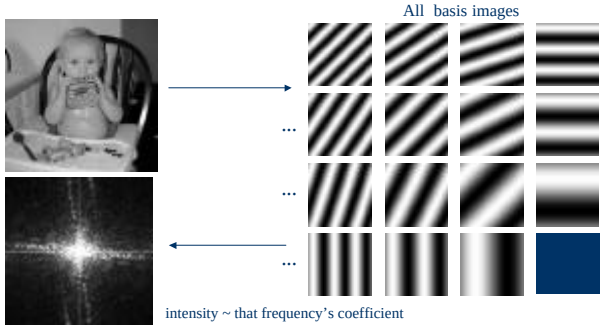


- What does the F-t express?

Periodicity

- The fourier transform highlights/identifies periodic structure in images.
- This is seen as peaks (usually several) along the same axis as the repetition in the original image

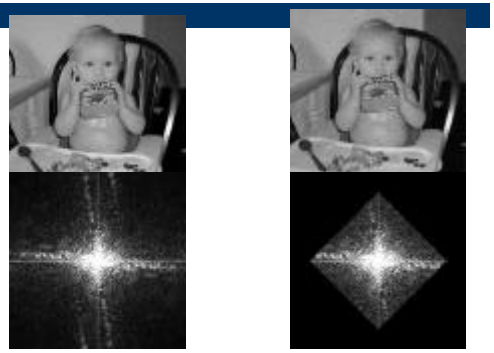
What does the F-t express?



Frequency spectrum

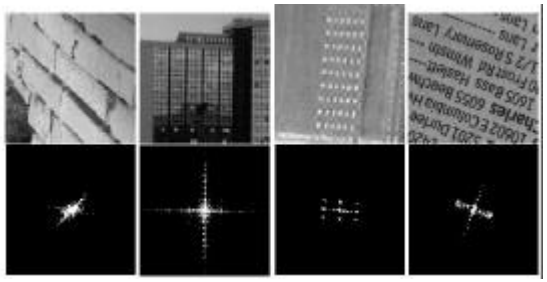
- By convention the lowest frequency components are shifted to the center of the image. (use fftshift in matlab)
- Fourier transform coefficients tells us the relative composition of low and high spatial frequencies, I.e. the frequency spectrum
- Usually more lows than highs

Cropping in Fourier space



Data Reduction: only use *some* of the existing frequencies

Properties: Rotation



- Notice direction of F-t:

Properties: Rotation

- Rotating f(x,y) by an angle θ_0 rotates F(u,v) by the same angle
- expressing the variables in polar coordinates

$$x = r\cos\theta,\ y = r\sin\theta,\ u = w\cos\phi,\ v = w\sin\phi$$

- substitution leads to

$$f(r,\theta+\theta_0) \leftrightarrow F(w,\phi+\theta_0)$$

Properties: Translation

$$f(x,y)e^{j2\pi(u_0x+v_0y)/N} \leftrightarrow F(u-u_0,v-v_0)$$

$$f(x-x_0,y-y_0) \leftrightarrow F(u,v)e^{-j2\pi(ux_0+vy_0)/N}$$

- the magnitude is unaffected because

$$\left|F(u,v)e^{-j2\pi(ux_0+vy_0)/N}\right| = \left|F(u,v)\right|$$

Properties: Distributivity

- the Fourier transform and its inverse are distributive over addition but not over multiplication:

$$\mathrm{F}\{f_1(x,y)+f_2(x,y)\} = \mathrm{F}\{f_1(x,y)\}+\mathrm{F}\{f_2(x,y)\}$$

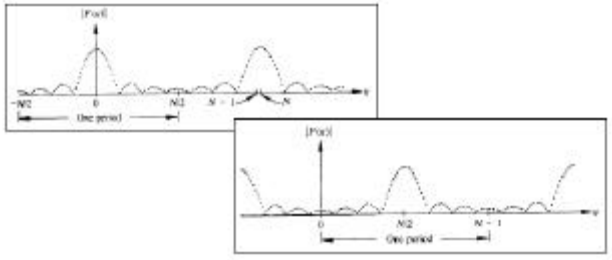
- and

$$\mathrm{F}\{f_1(x,y)*f_2(x,y)\} \neq \mathrm{F}\{f_1(x,y)\}*\mathrm{F}\{f_2(x,y)\}$$

Properties: Periodicity

- discrete Fourier transform and inverse are periodic with period N

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



Slow Fourier transform: DFT in matrix form

$$\begin{bmatrix} F(0) \\ F(2) \\ \vdots \\ F(N-1) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-\frac{j2\pi}{N}} & \cdots & e^{-\frac{j2\pi(N-1)}{N}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-\frac{j2\pi(N-1)}{N}} & \cdots & e^{-\frac{j2\pi(N-1)^2}{N}} \end{bmatrix} \begin{bmatrix} f(0) \\ f(2) \\ \vdots \\ f(N-1) \end{bmatrix}$$

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-\frac{j2\pi ux}{N}}$$

- Note: u=Ax form!
- O(n^2) due to matrix mult

Inverse: A^{-1}

- y

Fast Fourier Transform, FFT

- Recall DFT:

$$F(u) = \sum_{x=0}^{N-1} f(x) e^{-j\frac{2}{N}\pi ux}$$

$$= \sum_{x=0}^{N/2-1} f(2x) e^{-j\frac{2}{N}\pi u2x} + \sum_{x=0}^{N/2-1} f(2x+1) e^{-j\frac{2}{N}\pi u(2x+1)}$$

$$= \sum_{x=0}^{N/2-1} f(2x) e^{-j\frac{2}{N/2}\pi ux} + e^{-j\frac{2}{N}\pi u} \sum_{x=0}^{N/2-1} f(2x+1) e^{-j\frac{2}{N/2}\pi ux}$$

$$= F^{Even} + e^{-j\frac{2}{N}\pi u} F^{Odd}$$

DFT vs FFT time comparison

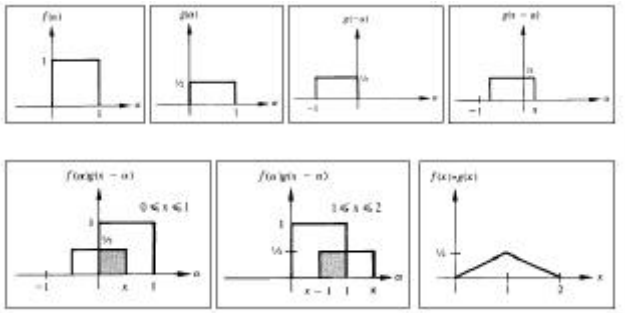
- DFT: F=Af
- O(n^2) time complexity

- FFT: F(0..255) = F(0,2,…,254)+eF(1,3,…,255)
- So F(256)->2xF(128)->4xF(64)->…->256xF(1)
- In general: O(nlogn) time complexity

Convolution

- Like an inner product, but with g reversed.
- Def: $f(x) \tilde{a} g(x) = \int_{-\infty}^{\infty} f(\ddot{e}) g(x - \ddot{e}) d\ddot{e}$ (Gonzalez 3.3.23)
- Theorem: $=(f \tilde{a} g) = FG$
- Convolution integral -> Point wise multiplication!!
- Discrete: FFT+Convolution theorem = fast convolutions.

Convolution example



Discrete convolution

- Gonzalez 3.3.29

$$f(x) \tilde{a} g(x) = \sum_{\ddot{e}=0}^{M-1} f(\ddot{e}) g(x - \ddot{e})$$

- Can also be written with a circulant matrix, e.g.:

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tilde{a} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

Convolution: Applications

- Models many physical processes like lens and diffraction distortion/degradation.
- Can also be used to “undo” degratation in image enhancement.
- Basic operation in shift invariant linear systems.
- (Used to describe generalized functions)

Practical discrete FFT-convolution

1. Let x=[1,2,3,2,1], y=[1,1,1]
2. Pad to: x=[0,1,2,3,2,1,0,0], y=[0,0,1,1,1,0,0,0]
3. Compute FFT-> X, Y
4. Compute pointwise product-> XY=Z
5. Compute IFFT(Z)

(Result hopefully =[1,3,6,7,6,3,1,0])

- Note need n=2^k for FFT
- Pad with zeros to avoid overlap and get 2^k