CS534: Introduction to Computer Vision
Fourier Transform

Ahmed Elgammal
Dept. of Computer Science
Rutgers University

Outlines

- Fourier Series and Fourier integral
- Fourier Transform (FT)
- Discrete Fourier Transform (DFT)
- Aliasing and Nyquest Theorem
- 2D FT and 2D DFT
- Application of 2D-DFT in imaging
- Inverse Convolution
- Discrete Cosine Transform (DCT)

Sources:

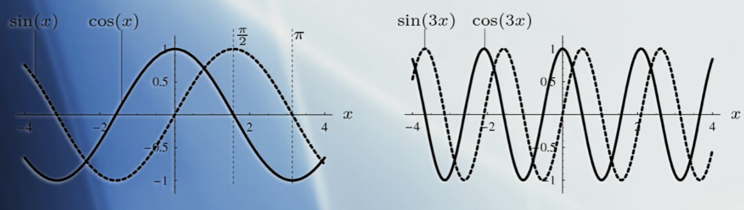
- Forsyth and Ponce, Chapter 7
- Burger and Burge "Digital Image Processing" Chapter 13, 14, 15
- Fourier transform images from Prof. John M. Brayer @ UNM

http://www.cs.unm.edu/~brayer/vision/fourier.html

Basics

- Sine and Cosine functions are periodic
  $\cos(x) = \cos(x + 2\pi) = \cos(x + 4\pi) = \cdots = \cos(x + k2\pi)$
- Angular Frequency (ω) and Amplitude (a)
  $a \cdot \cos(\omega x)$ and $a \cdot \sin(\omega x)$
- Angular Frequency (ω) : number of oscillations over the distance 2π
  T: the time for a complete cycle
  $$T = \frac{2\pi}{\omega}$$
- Common Frequency f: number of oscillation in a unit time

$$f = \frac{1}{T} = \frac{\omega}{2\pi} \quad \text{or} \quad \omega = 2\pi f$$



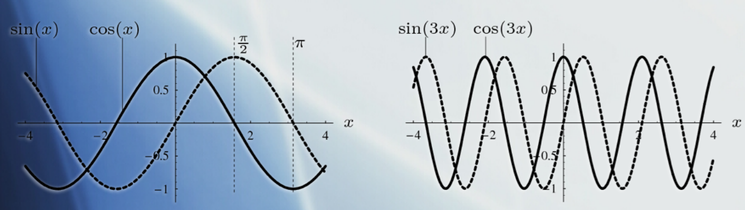
(a) (b)

Basics

- Phase: Shifting a cosine function along the x axis by a distance φ change the phase of the cosine wave. φ denotes the phase angle

$$\cos(x) \rightarrow \cos(x - \varphi)$$

$$\sin(\omega x) = \cos\left(\omega x - \frac{\pi}{2}\right)$$

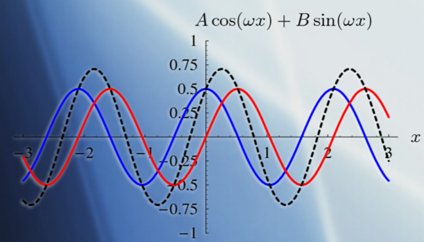


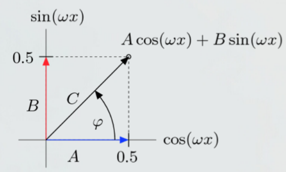
(a) (b)

- Adding cosines and sines with the same frequency results in another sinusoid

$$A \cdot \cos(\omega x) + B \cdot \sin(\omega x) = C \cdot \cos(\omega x - \varphi)$$

$C = \sqrt{A^2 + B^2}$ and $\varphi = \tan^{-1}\left(\frac{B}{A}\right)$





$$e^{i\theta} = e^{i\omega x} = \cos(\omega x) + i \cdot \sin(\omega x)$$

Fourier Series and Fourier integral

- We can represent any periodic function as sum of pairs of sinusoidal functions- using a basic (fundamental) frequency

$$g(x) = \sum_{k=0}^{\infty} [A_k \cos(k\omega_0 x) + B_k \sin(k\omega_0 x)]$$

- Fourier Integral: any function can be represented as combination of sinusoidal functions with infinitely many frequencies

$$g(x) = \int_0^{\infty} A_\omega \cos(\omega x) + B_\omega \sin(\omega x)\, d\omega$$

- Fourier Integral

$$g(x) = \int_0^\infty A_\omega \cos(\omega x) + B_\omega \sin(\omega x) \,\mathrm{d}\omega$$

- How much of each frequency contributes to a given function

$$A_\omega = A(\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} g(x) \cdot \cos(\omega x) \,\mathrm{d}x$$

$$B_\omega = B(\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} g(x) \cdot \sin(\omega x) \,\mathrm{d}x$$

Fourier Transform

$$e^{\mathrm{i}\theta} = e^{\mathrm{i}\omega x} = \cos(\omega x) + \mathrm{i} \cdot \sin(\omega x)$$

$$A_\omega = A(\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} g(x) \cdot \cos(\omega x) \,\mathrm{d}x$$

$$B_\omega = B(\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} g(x) \cdot \sin(\omega x) \,\mathrm{d}x$$

$$\begin{aligned} G(\omega) &= \sqrt{\tfrac{\pi}{2}} \Big[A(\omega) - \mathrm{i} \cdot B(\omega) \Big] \\ &= \sqrt{\tfrac{\pi}{2}} \left[\frac{1}{\pi} \int_{-\infty}^{\infty} g(x) \cdot \cos(\omega x) \,\mathrm{d}x \;-\; \mathrm{i} \cdot \frac{1}{\pi} \int_{-\infty}^{\infty} g(x) \cdot \sin(\omega x) \,\mathrm{d}x \right] \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(x) \cdot \Big[\cos(\omega x) - \mathrm{i} \cdot \sin(\omega x) \Big] \,\mathrm{d}x, \end{aligned}$$

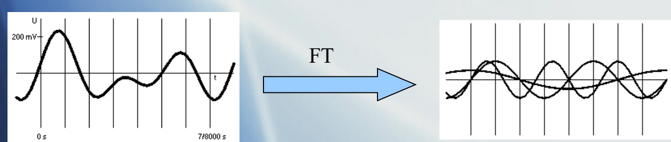
$$\boxed{\begin{aligned} G(\omega) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(x) \cdot \Big[\cos(\omega x) - \mathrm{i} \cdot \sin(\omega x) \Big] \,\mathrm{d}x \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(x) \cdot e^{-\mathrm{i}\omega x} \,\mathrm{d}x. \end{aligned}}$$

- Fourier transform

$$\begin{aligned} G(\omega) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(x) \cdot \left[\cos(\omega x) - \mathrm{i} \cdot \sin(\omega x)\right] \, \mathrm{d}x \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(x) \cdot e^{-\mathrm{i}\omega x} \, \mathrm{d}x. \end{aligned}$$

- Inverse Fourier transform

$$\begin{aligned} g(x) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} G(\omega) \cdot \left[\cos(\omega x) + \mathrm{i} \cdot \sin(\omega x)\right] \, \mathrm{d}\omega \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} G(\omega) \cdot e^{\mathrm{i}\omega x} \, \mathrm{d}\omega. \end{aligned}$$



Frequency domain

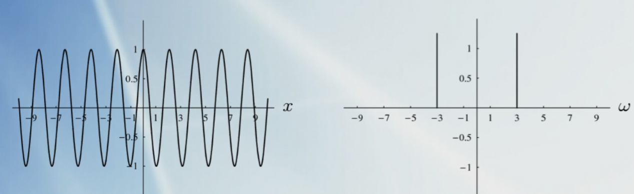
Temporal or spatial domain

Fourier Transform

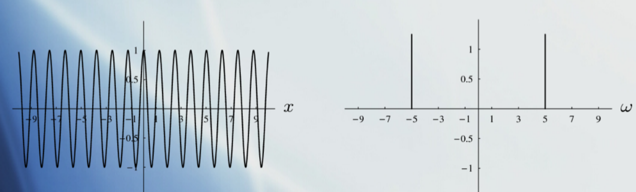
- The forward and inverse transformation are almost similar (only the sign in the exponent is different)
- any signal is represented in the frequency space by its frequency "spectrum"
- The Fourier spectrum is uniquely defined for a given function. The opposite is also true.
- Fourier transform pairs

$$g(x) \circ\!\!-\!\!\bullet\; G(\omega)$$

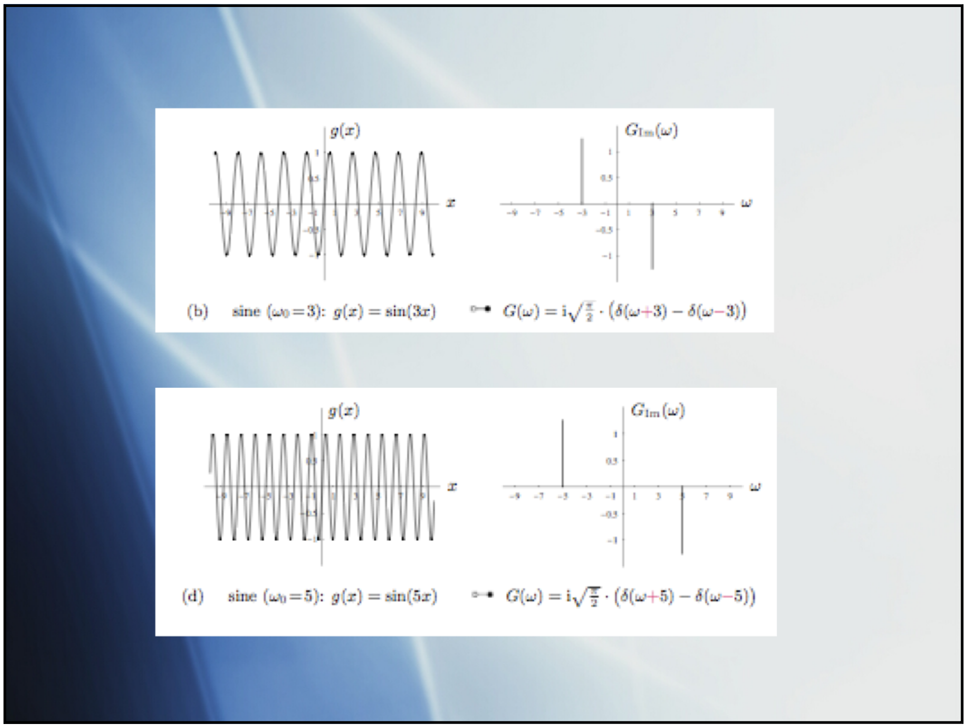
Function	*Transform Pair* $g(x) \circ\!\!-\!\!\bullet\, G(\omega)$	*Figure*
Cosine function with frequency ω_0	$g(x) = \cos(\omega_0 x)$ $G(\omega) = \sqrt{\frac{\pi}{2}} \cdot \left(\delta(\omega - \omega_0) + \delta(\omega + \omega_0)\right)$	13.3 (a, c)
Sine function with frequency ω_0	$g(x) = \sin(\omega_0 x)$ $G(\omega) = \mathrm{i}\sqrt{\frac{\pi}{2}} \cdot \left(\delta(\omega - \omega_0) - \delta(\omega + \omega_0)\right)$	13.3 (b, d)
Gaussian function of width σ	$g(x) = \frac{1}{\sigma} \cdot e^{-\frac{x^2}{2\sigma^2}}$ $G(\omega) = e^{-\frac{\sigma^2 \omega^2}{2}}$	13.4 (a, b)
Rectangular pulse of width $2b$	$g(x) = \Pi_b(x) = \begin{cases} 1 & \text{for } \lvert x \rvert \le b \\ 0 & \text{otherwise} \end{cases}$ $G(\omega) = \frac{2b \sin(b\omega)}{\sqrt{2\pi}\omega}$	13.4 (c, d)



(a) cosine ($\omega_0 = 3$): $g(x) = \cos(3x)$ $\circ\!\!-\!\!\bullet$ $G(\omega) = \sqrt{\frac{\pi}{2}} \cdot \left(\delta(\omega - 3) + \delta(\omega + 3)\right)$

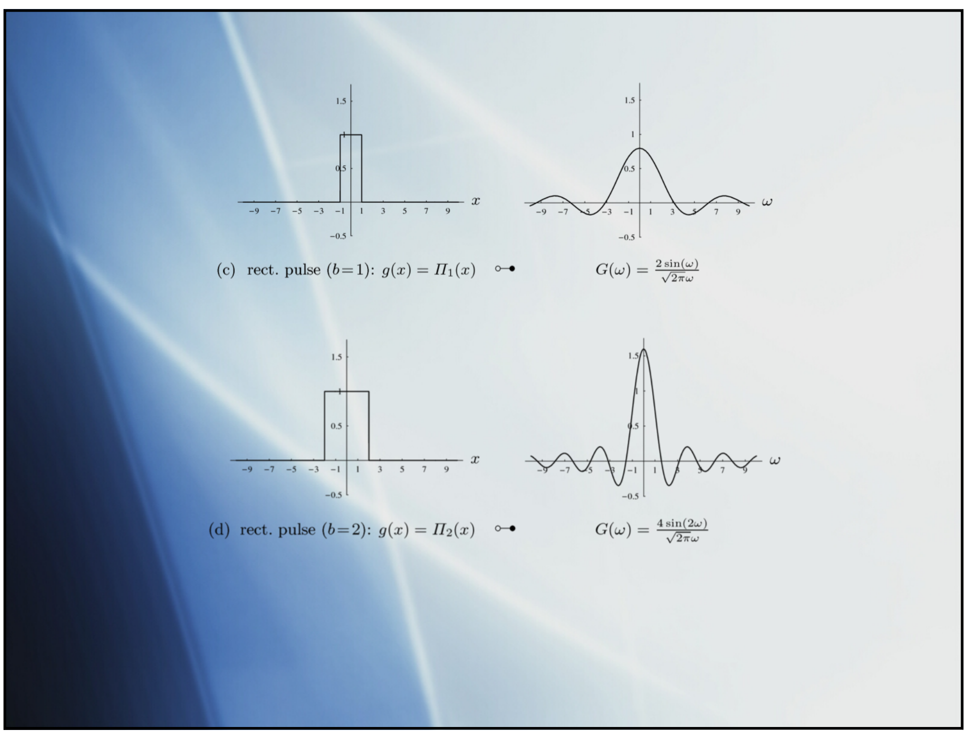


(c) cosine ($\omega_0 = 5$): $g(x) = \cos(5x)$ $\circ\!\!-\!\!\bullet$ $G(\omega) = \sqrt{\frac{\pi}{2}} \cdot \left(\delta(\omega - 5) + \delta(\omega + 5)\right)$



(b) sine ($\omega_0=3$): $g(x)=\sin(3x)$ ⊶ $G(\omega)=\mathrm{i}\sqrt{\frac{\pi}{2}}\cdot(\delta(\omega+3)-\delta(\omega-3))$

(d) sine ($\omega_0=5$): $g(x)=\sin(5x)$ ⊶ $G(\omega)=\mathrm{i}\sqrt{\frac{\pi}{2}}\cdot(\delta(\omega+5)-\delta(\omega-5))$

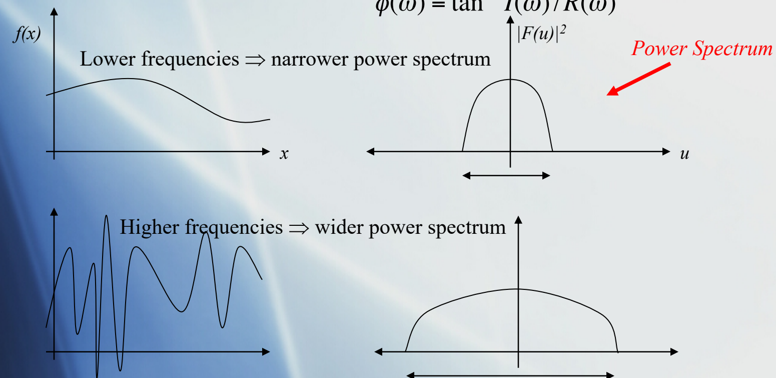


(c) rect. pulse ($b=1$): $g(x)=\Pi_1(x)$ ⊶ $G(\omega)=\frac{2\sin(\omega)}{\sqrt{2\pi}\omega}$

(d) rect. pulse ($b=2$): $g(x)=\Pi_2(x)$ ⊶ $G(\omega)=\frac{4\sin(2\omega)}{\sqrt{2\pi}\omega}$

- Since of FT of a real function is generally complex, we use magnitude and phase

$$G(\omega) = R(\omega) + jI(\omega) \Rightarrow \begin{aligned} |G(\omega)| &= \left(R^2(\omega) + I^2(\omega)\right)^{1/2} \\ \phi(\omega) &= \tan^{-1} I(\omega)/R(\omega) \end{aligned}$$



Properties

- Symmetry: for real-valued functions

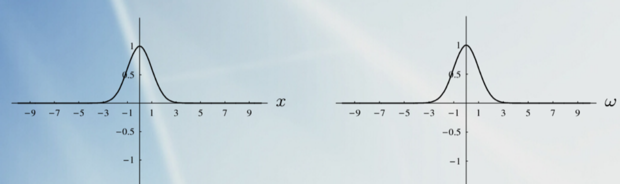
$$G(\omega) = G^*(-\omega)$$

- Linearity

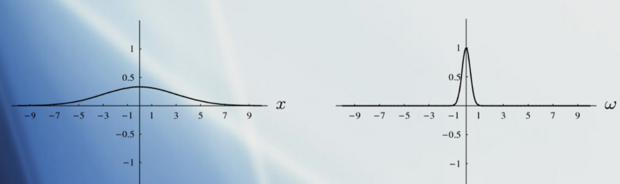
$$c \cdot g(x) \circ\!\!-\!\!\bullet\ c \cdot G(\omega)$$
$$g_1(x) + g_2(x) \circ\!\!-\!\!\bullet\ G_1(\omega) + G_2(\omega)$$

- Similarity

$$g(sx) \circ\!\!-\!\!\bullet \frac{1}{|s|} \cdot G\left(\frac{\omega}{s}\right)$$



(a) Gauss ($\sigma = 1$): $g(x) = e^{-\frac{x^2}{2}}$ $\circ\!\!-\!\!\bullet$ $G(\omega) = e^{-\frac{\omega^2}{2}}$



(b) Gauss ($\sigma = 3$): $g(x) = \frac{1}{3} \cdot e^{-\frac{x^2}{2 \cdot 9}}$ $\circ\!\!-\!\!\bullet$ $G(\omega) = e^{-\frac{9\omega^2}{2}}$

Important Properties:

- FT and Convolution
- Convolving two signals is equivalent to multiplying their Fourier spectra

$$g(x) * h(x) \circ\!\!-\!\!\bullet\; G(\omega) \cdot H(\omega)$$

- Multiplying two signals is equivalent to convolving their Fourier spectra

$$g(x) \cdot h(x) \circ\!\!-\!\!\bullet\; G(\omega) * H(\omega)$$

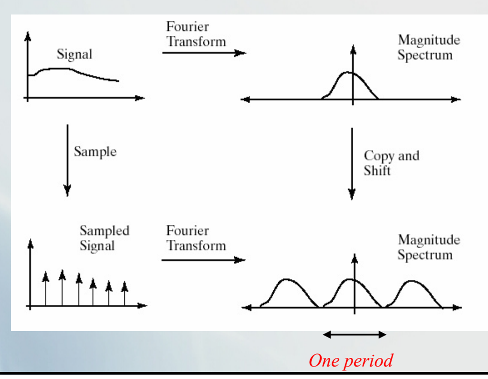
- FT of a Gaussian with sd=σ is a Gaussian with sd=1/σ

Fourier Transform of discrete signals

- If we discretize *f(x)* using uniformly spaced samples *f(0), f(1),…,f(N-1)*, we can obtain FT of the sampled function

- Important Property:

Periodicity *F(m)=F(m+N)*



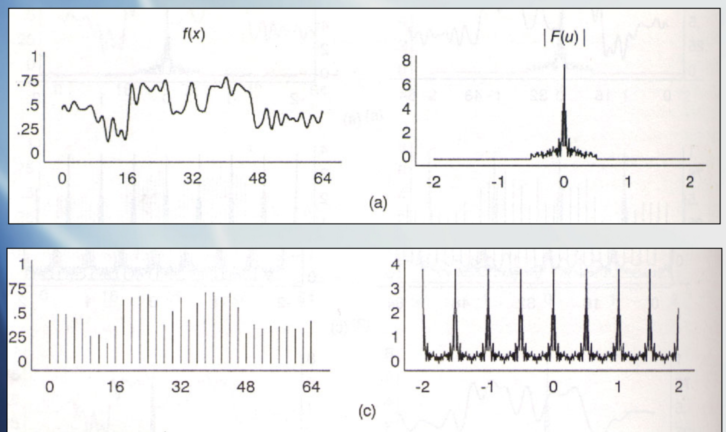
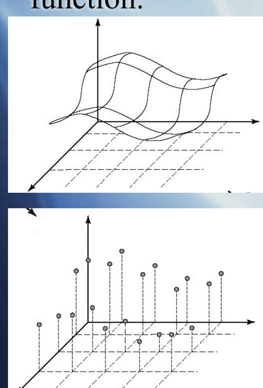


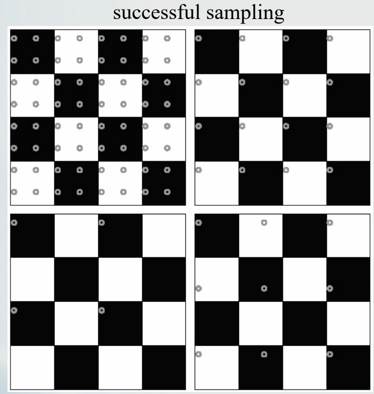
Image from *Computer Graphics: Principles and Practice* by Foley, van Dam, Feiner, and Hughes

Sampling and Aliasing

- Differences between continuous and discrete images
- Images are sampled version of a continuous brightness function.



successful sampling

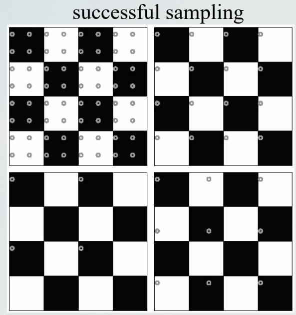


unsuccessful sampling

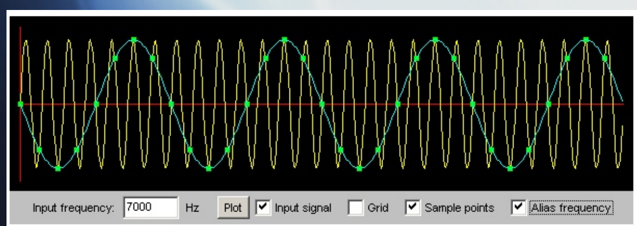
Sampling and Aliasing

- Sampling involves loss of information
- Aliasing: high spatial frequency components appear as low spatial frequency components in the sampled signal

successful sampling



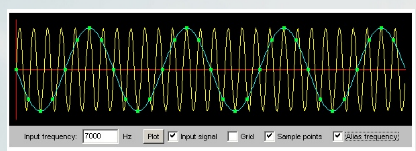
unsuccessful sampling

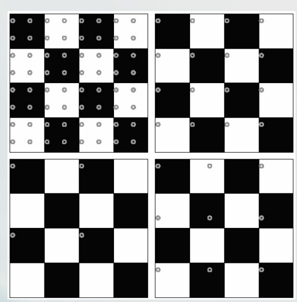


Java applet from: http://www.dsptutor.freeuk.com/aliasing/AD102.html

Aliasing

- *Nyquist theorem*: The sampling frequency must be at least twice the highest frequency present for a signal to be reconstructed from a sampled version. *(Nyquist frequency)*





Temporal Aliasing

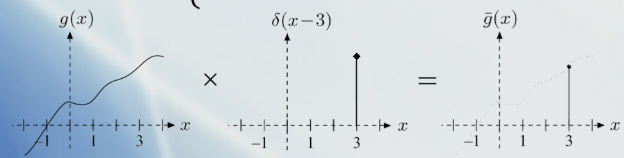
The wheel appears to be moving backwards at about ¼ angular frequency

Understanding the Sampling Process - Impulse function

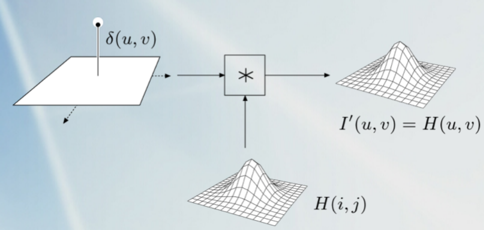
$$\delta(x) = 0 \text{ for } x \neq 0 \quad \text{and} \quad \int_{-\infty}^{\infty} \delta(x)\, \mathrm{d}x = 1$$

$$\delta(sx) = \frac{1}{|s|} \cdot \delta(x) \quad \text{for } s \neq 0$$

$$\bar{g}(x) = g(x) \cdot \delta(x - x_0) = \begin{cases} g(x_0) & \text{for } x = x_0 \\ 0 & \text{otherwise} \end{cases}$$

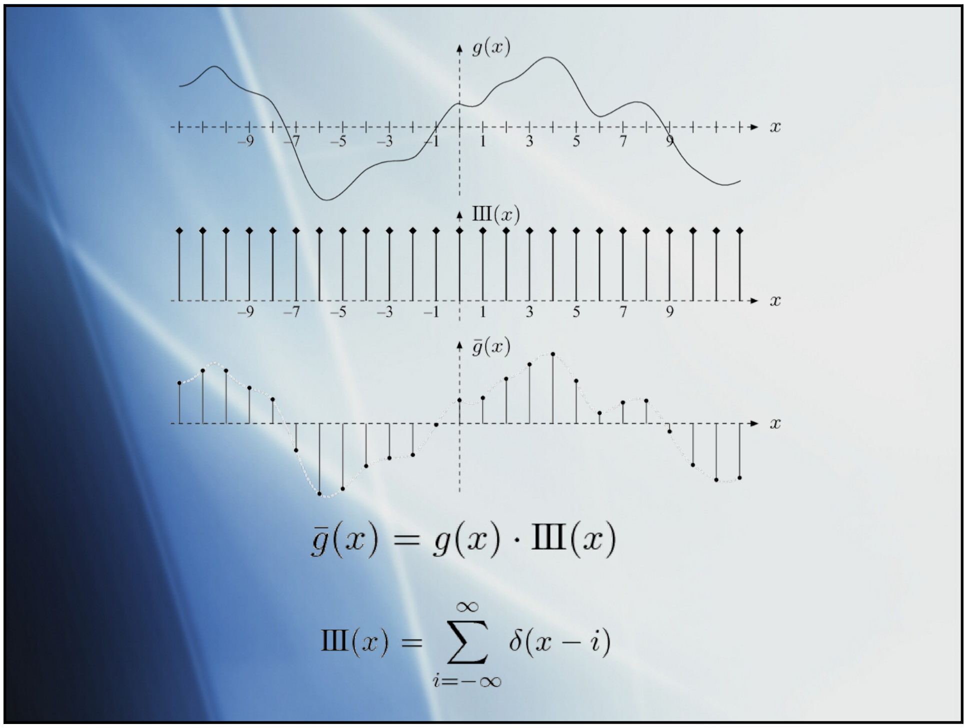


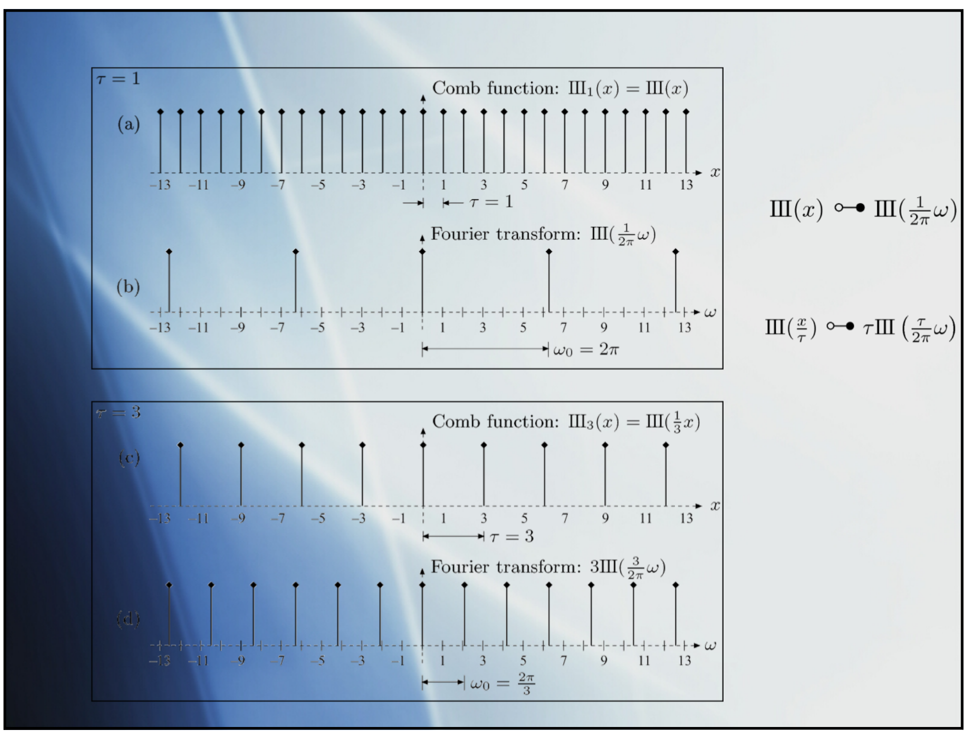
$$f(x) * \delta(x-d) = f(x-d)$$

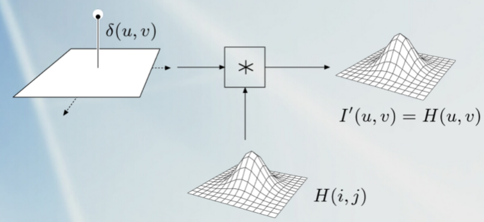


$$f(x) * \delta(x-d) = f(x-d)$$

Convolving a function with an impulse at d, shifts the function to d


$g(x)$
x
$\text{III}(x)$
$\bar{g}(x)$
$\bar{g}(x) = g(x) \cdot \text{III}(x)$
$\text{III}(x) = \sum_{i=-\infty}^{\infty} \delta(x-i)$


$\tau = 1$
Comb function: $\text{III}_1(x) = \text{III}(x)$
(a)
$\tau = 1$
Fourier transform: $\text{III}(\frac{1}{2\pi}\omega)$
(b)
$\omega_0 = 2\pi$
$\tau = 3$
Comb function: $\text{III}_3(x) = \text{III}(\frac{1}{3}x)$
(c)
$\tau = 3$
Fourier transform: $3\text{III}(\frac{3}{2\pi}\omega)$
(d)
$\omega_0 = \frac{2\pi}{3}$
$\text{III}(x) \circ\!\!-\!\!\bullet\ \text{III}(\frac{1}{2\pi}\omega)$
$\text{III}(\frac{x}{\tau}) \circ\!\!-\!\!\bullet\ \tau\text{III}\left(\frac{\tau}{2\pi}\omega\right)$

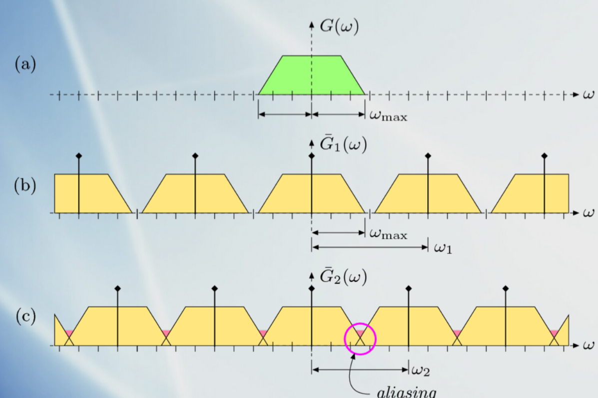


$$f(x) * \delta(x-d) = f(x-d)$$

$$\text{III}(x) = \sum_{i=-\infty}^{\infty} \delta(x-i)$$

$$f(x) * \text{III}(x) = \sum_{-\infty}^{\infty} f(x-i)$$

Convolving a function with an impulse at d, shifts the function to d
Convolving a function *f* with a comp function copies *f* at each impulse

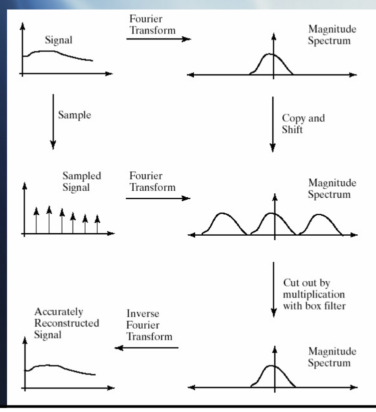


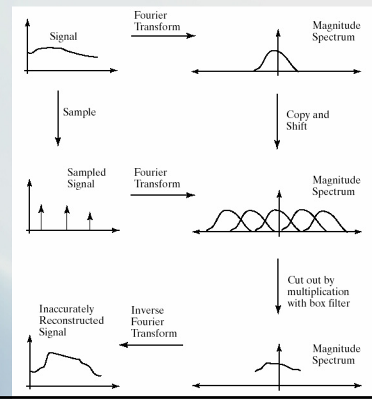
$$g(x) \cdot \text{III}(\tfrac{x}{\tau}) \multimap G(\omega) * \tau\text{III}\left(\tfrac{\tau}{2\pi}\omega\right)$$

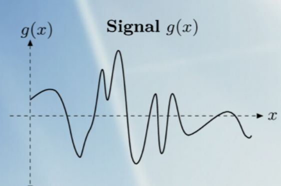
$$\omega_{\max} \le \frac{1}{2}\omega_s \quad \text{or} \quad \omega_s \ge 2\omega_{\max}$$

Sampling, aliasing, and DFT

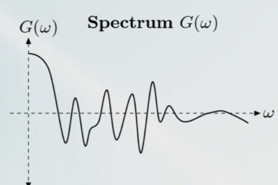
- DFT consists of a sum of copies of the FT of the original signal shifted by the sampling frequency:
  - If shifted copies do not intersect: reconstruction is possible.
  - If shifted copies do intersect: incorrect reconstruction, high frequencies are lost (Aliasing)



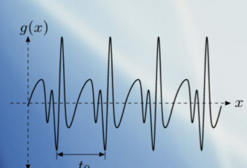




(a) Continuous nonperiodic signal.



(b) Continuous nonperiodic spectrum.

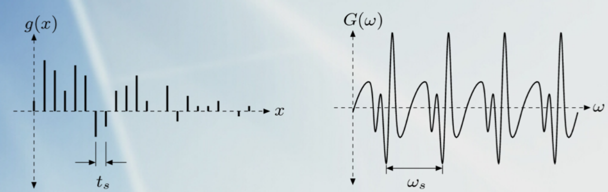


(c) Continuous periodic signal with period t_0.

(d) Discrete nonperiodic spectrum with values spaced at $\omega_0 = 2\pi/t_0$.

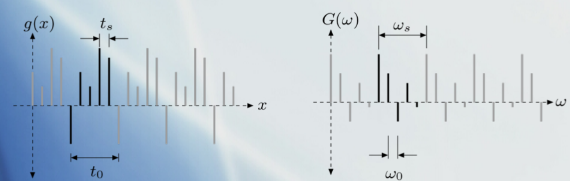
Recall Fourier series for periodic functions

$$g(x) = \sum_{k=0}^{\infty} [A_k \cos(k\omega_0 x) + B_k \sin(k\omega_0 x)]$$



(e) Discrete nonperiodic signal with samples spaced at t_s.

(f) Continuous periodic spectrum with period $\omega_s = 2\pi/t_s$.



(g) Discrete periodic signal with samples spaced at t_s and period $t_0 = t_s M$.

(h) Discrete periodic spectrum with values spaced at $\omega_0 = 2\pi/t_0$ and period $\omega_s = 2\pi/t_s = \omega_0 M$.

2-dimension

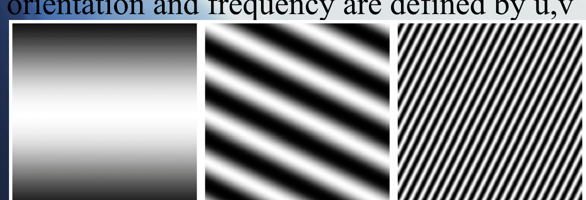
In two dimension

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

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

$$e^{-j2\pi(ux+vy)} = \cos(2\pi(ux+vy)) + j\sin(2\pi(ux+vy))$$

- These terms are sinusoids on the x,y plane whose orientation and frequency are defined by u,v



DFT in 2D

- For a 2D periodic function of size MxN, DFT is defined as:

$$\begin{aligned} G(m,n) &= \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} g(u,v) \cdot e^{-i2\pi \frac{mu}{M}} \cdot e^{-i2\pi \frac{nv}{N}} \\ &= \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} g(u,v) \cdot e^{-i2\pi (\frac{mu}{M} + \frac{nv}{N})} \end{aligned}$$

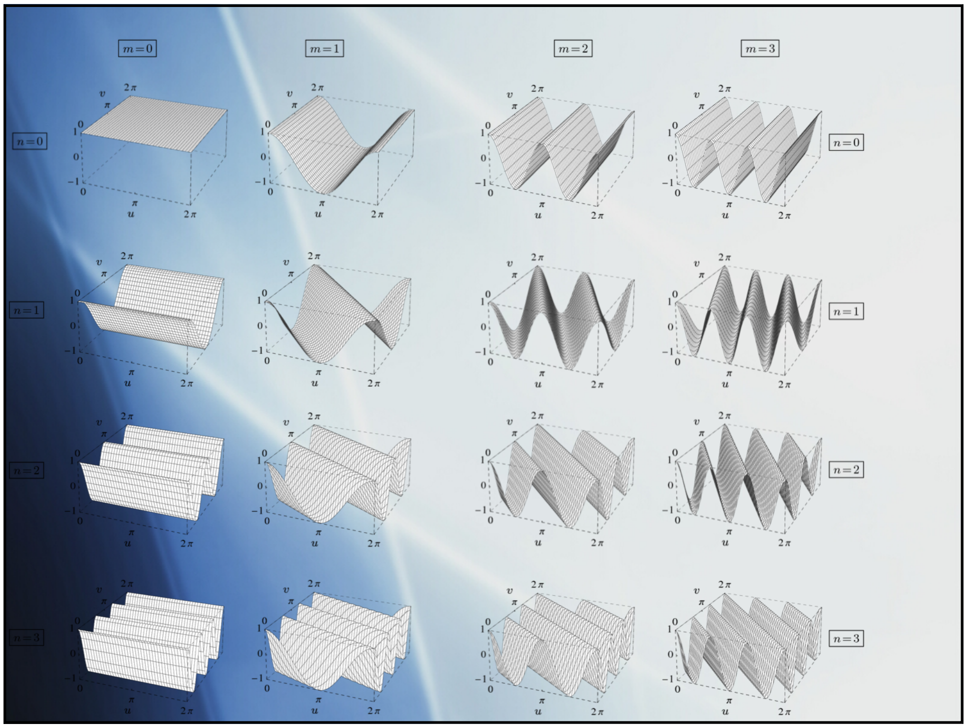
- Inverse transform

$$\begin{aligned} g(u,v) &= \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} G(m,n) \cdot e^{i2\pi \frac{mu}{M}} \cdot e^{i2\pi \frac{nv}{N}} \\ &= \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} G(m,n) \cdot e^{i2\pi (\frac{mu}{M} + \frac{nv}{N})} \end{aligned}$$

$$\begin{aligned} e^{i2\pi(\frac{mu}{M}+\frac{nv}{N})} &= e^{i(\omega_m u + \omega_n v)} \\ &= \underbrace{\cos\left[2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)\right]}_{\boldsymbol{C}_{m,n}^{M,N}(u,v)} + i \cdot \underbrace{\sin\left[2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)\right]}_{\boldsymbol{S}_{m,n}^{M,N}(u,v)} \end{aligned}$$

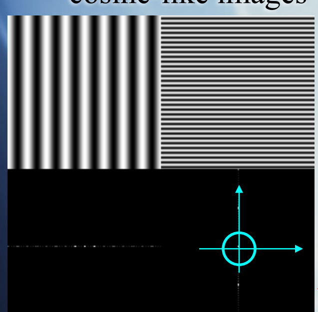
$$\boldsymbol{C}_{m,n}^{M,N}(u,v) = \cos\left[2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)\right] = \cos(\omega_m u + \omega_n v)$$

$$\boldsymbol{S}_{m,n}^{M,N}(u,v) = \sin\left[2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)\right] = \sin(\omega_m u + \omega_n v)$$



Visualizing 2D-DFT

- The FT tries to represent all images as a summation of cosine-like images



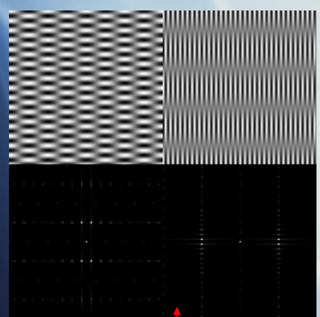
Images of pure cosines

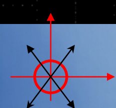
- *Center of the image:* the origin of the frequency coordinate system
- *m*-axis: (left to right) the horizontal component of frequency
- *n*-axis: (bottom-top) the vertical component of frequency
- Center dot (0,0) frequency : image average

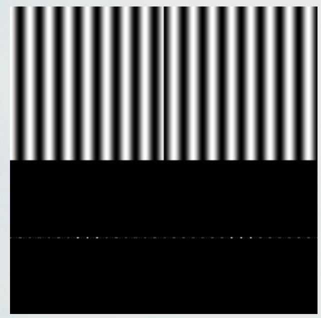
FT

- high frequencies in the vertical direction will cause bright dots away from the center in the vertical direction.
- high frequencies in the horizontal direction will cause bright dots away from the center in the horizontal direction.

- Since images are real numbers (not complex) FT image is symmetric around the origin.



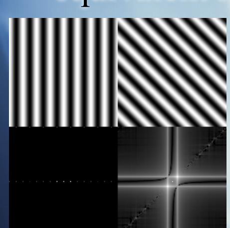


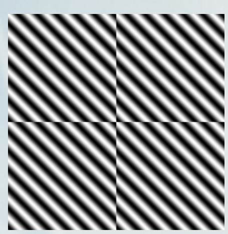


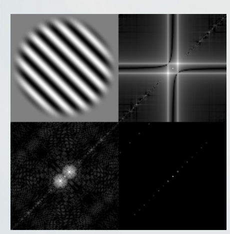
FT: symmetry

FT is shift invariant

- In general, rotation of the image results in equivalent rotation of its FT



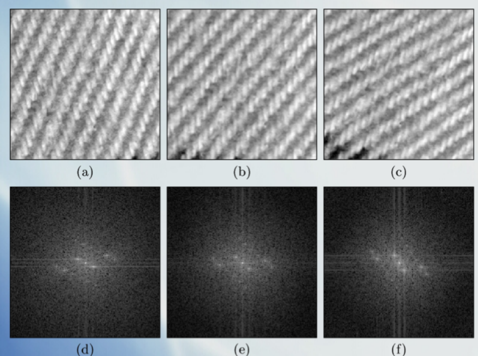




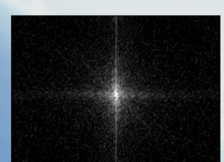
Why it is not the case ?

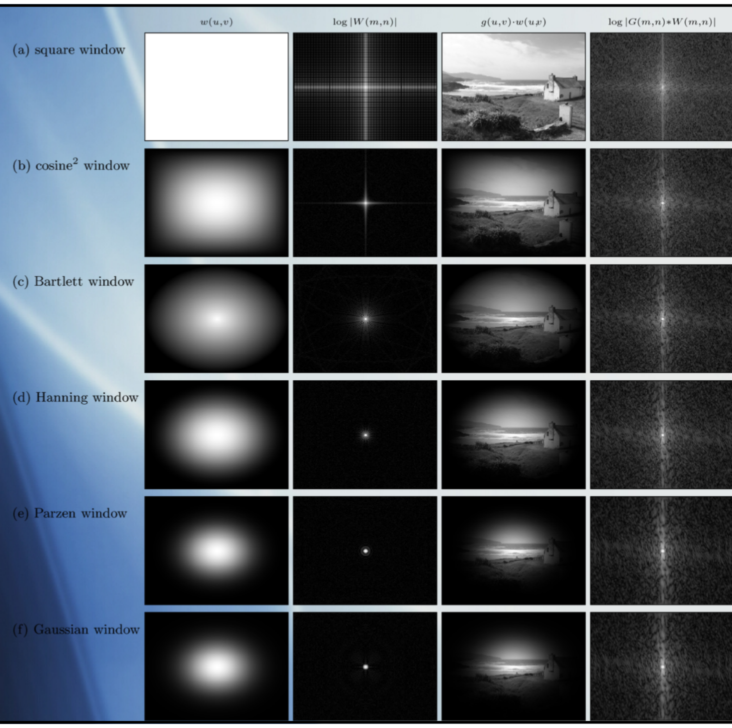
- Edge effect !
- FT always treats an image as if it were part of a periodically replicated array of identical images extending horizontally and vertically to infinity
- Solution: “windowing” the image

Rotation

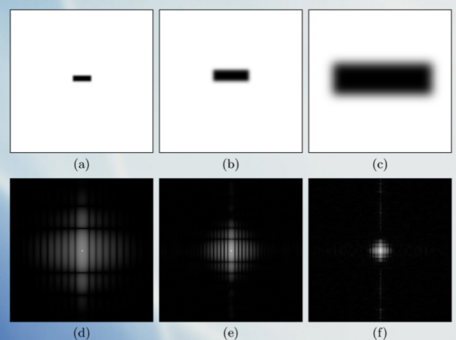


Edge effect

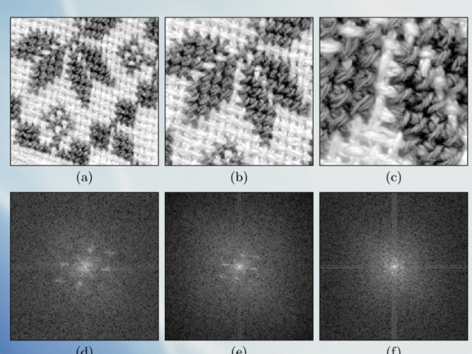



w(u,v)
log |W(m,n)|
g(u,v)·w(u,v)
log |G(m,n)∗W(m,n)|
(a) square window
(b) cosine² window
(c) Bartlett window
(d) Hanning window
(e) Parzen window
(f) Gaussian window

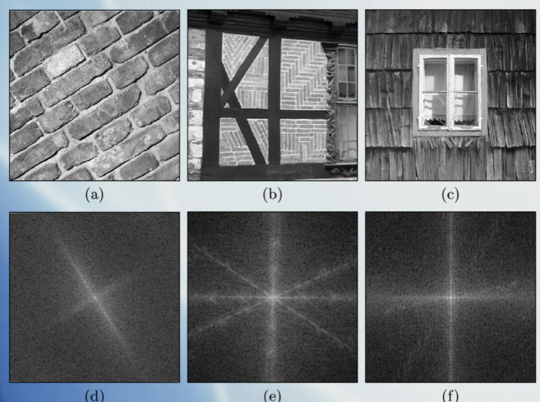
Scaling

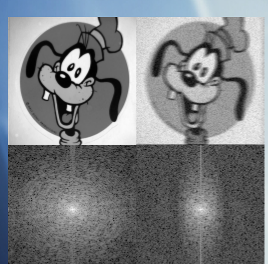


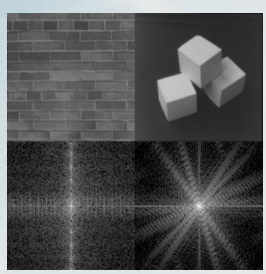
Periodic image patterns



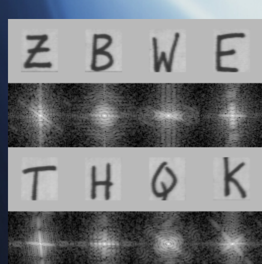
Oriented, elongated structures

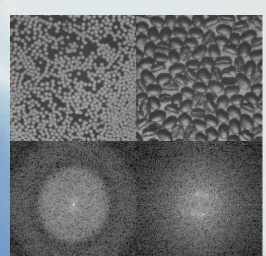


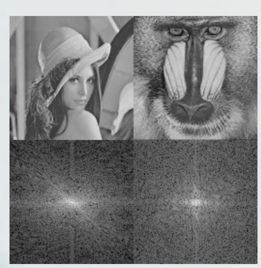




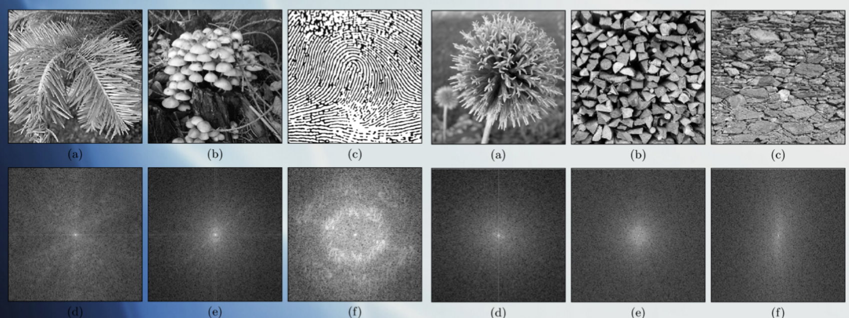
- notice a bright band going to high frequencies perpendicular to the strong edges in the image
- Anytime an image has a strong-contrast, sharp edge the gray values must change very rapidly. It takes lots of high frequency power to follow such an edge so there is usually such a line in its magnitude spectrum.



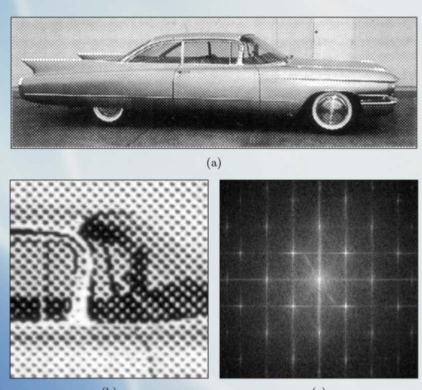




Natural Images



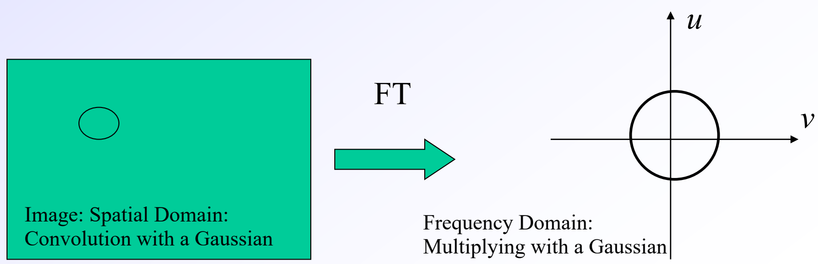
Print patterns



Gaussian Filter and Smoothing

Gaussian Filter is Low-Pass Filter:

- Recall: Convolution in the image domain is equivalent to multiplication in the Frequency domain.
- Recall: FT of a Gaussian with sd=σ is a Gaussian with sd=$1/\sigma$
- Therefore, convolving an image with a Gaussian with sd=σ is equivalent to multiplying it's FT with a Gaussian with sd=$1/\sigma$
- Therefore we will get rid of high frequencies.
- Smoothing with a Gaussian with a very small $\sigma \Rightarrow$ get rid of highest spatial frequencies $\Rightarrow$ Gaussian is a low-pass filter



FT applications in imaging

- Accelerated convolution
- De-convolution (inverse convolution)
- Image enhancement
- Image and Video Compression
- Saliency detection (spatial and/or temporal)

Linear Filters in Frequency space

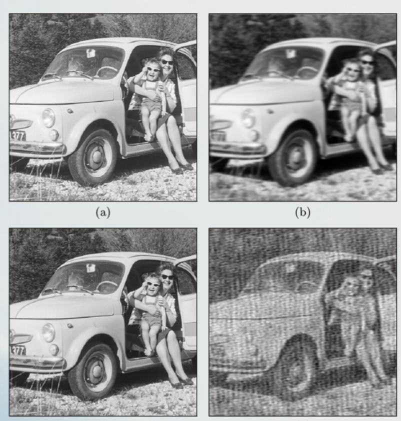
$$
\begin{array}{lccccc}
\text{Image space:} & g(u,v) & * & h(u,v) & = & g'(u,v) \\
 & \downarrow & & \downarrow & & \uparrow \\
 & \mathsf{DFT} & & \mathsf{DFT} & & \mathsf{DFT}^{-1} \\
 & \downarrow & & \downarrow & & \uparrow \\
\text{Frequency space:} & G(m,n) & \cdot & H(m,n) & \longrightarrow & G'(m,n)
\end{array}
$$

Inverse Filters - De-convolution

- How can we remove the effect of a filter ?

$$g_{\text{blur}} = g_{\text{orig}} * h_{\text{blur}}$$

$$G_{\text{blur}} = G_{\text{orig}} \cdot H_{\text{blur}}$$



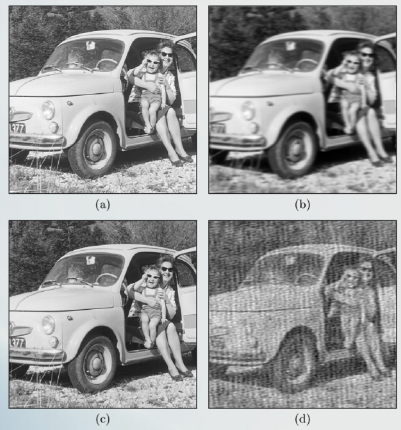
Inverse Filters - De-convolution

- How can we remove the effect of a filter ?

$$g_{\text{blur}} = g_{\text{orig}} * h_{\text{blur}}$$

$$G_{\text{blur}} = G_{\text{orig}} \cdot H_{\text{blur}}$$

$$G_{\text{orig}}(m,n) = \frac{G_{\text{blur}}(m,n)}{H_{\text{blur}}(m,n)}$$



Accelerated convolution in CNN

M Mathieu, Mikael Henaff and Yann LeCun
"Fast Training of Convolutional Networks through FFTs" 2014

In the forward pass, each output feature map is computed as a sum of the input feature maps convolved with the corresponding trainable weight kernel:

$$y_{f'} = \sum_f x_f * w_{f'f} \tag{1}$$

During the backward pass, the gradients with respect to the inputs are computed by convolving the transposed weight kernel with the gradients with respect to the outputs:

$$\frac{\partial L}{\partial x_f} = \frac{\partial L}{\partial y_{f'}} * w^T_{f'f} \tag{2}$$

This step is necessary for computing the gradients in (3) for the previous layer. Finally, the gradients of the loss with respect to the weight are computed by convolving each input feature map with the gradients with respect to the outputs:

$$\frac{\partial L}{w_{f'f}} = \frac{\partial L}{\partial y_{f'}} * x_f \tag{3}$$

Note that $\frac{\partial L}{\partial y_{f'}}$ is a 2-D matrix with the same dimensions as the output feature map $y_{f'}$, and that all operations consist of convolutions between various sets of 2-D matrices.

Accelerated convolution in CNN

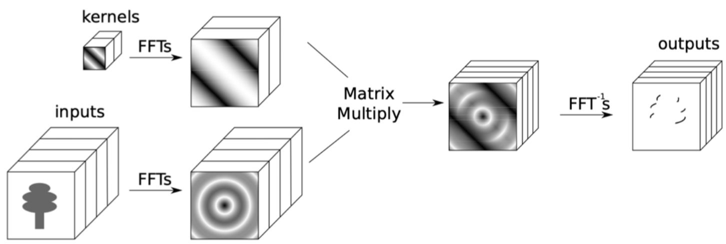


Figure 1: Illustration of the algorithm. Note that the matrix-multiplication involves multiplying all input feature maps by all corresponding kernels.

M Mathieu et al "Fast Training of Convolutional Networks through FFTs" 2014

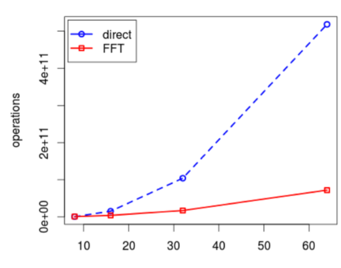
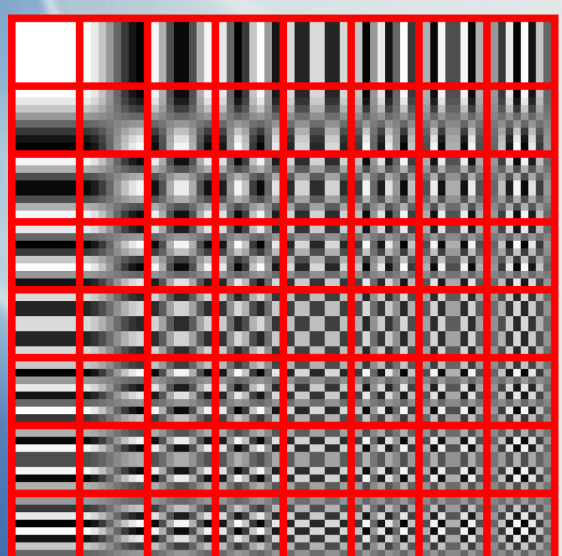
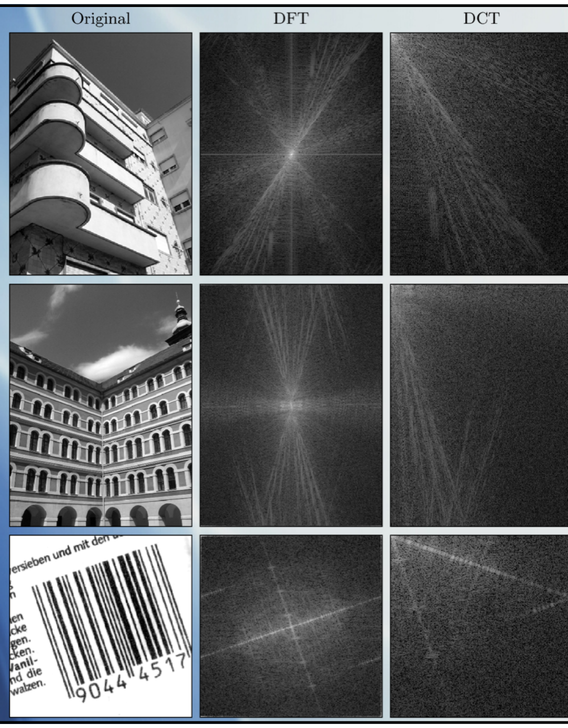


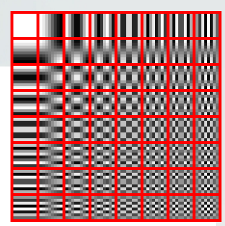
Figure 2: Number of operations required for computing (1) with different input image sizes and $S = 128, f = 96, f' = 256, k = 7$.

Image and Video Compression
The Discrete Cosine Transform (DCT)

- FT and DFT are designed for processing complex-valued signal and always produce a complex-valued spectrum.
- For a real-valued signal, the Fourier spectrum is symmetric
- Discrete Cosine Transform (DCT): similar to DFT but does not work with complex signals.
- DCT uses cosine functions only, with various wave numbers as the basis functions and operates on real-valued signals







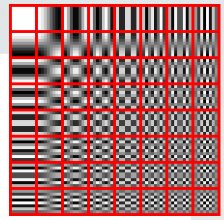
An 8×8 block from the Y image of 'Lena'

200	202	189	188	189	175	175	175
200	203	198	188	189	182	178	175
203	200	200	195	200	187	185	175
200	200	200	200	197	187	187	187
200	205	200	200	195	188	187	175
200	200	200	200	200	190	187	175
205	200	199	200	191	187	187	175
210	200	200	200	188	185	187	186

$f(i, j)$

515	65	-12	4	1	2	-8	5
-16	3	2	0	0	-11	-2	3
-12	6	11	-1	3	0	1	-2
-8	3	-4	2	-2	-3	-5	-2
0	-2	7	-5	4	0	-1	-4
0	-3	-1	0	4	1	-1	0
3	-2	-3	3	3	-1	-1	3
-2	5	-2	4	-2	2	-3	0

$F(u, v)$



Another 8 × 8 block from the Y image of 'Lena'

70	70	100	70	87	87	150	187
85	100	96	79	87	154	87	113
100	85	116	79	70	87	86	196
136	69	87	200	79	71	117	96
161	70	87	200	103	71	96	113
161	123	147	133	113	113	85	161
146	147	175	100	103	103	163	187
156	146	189	70	113	161	163	197

$f(i,j)$

-80	-40	89	-73	44	32	53	-3
-135	-59	-26	6	14	-3	-13	-28
47	-76	66	-3	-108	-78	33	59
-2	10	-18	0	33	11	-21	1
-1	-9	-22	8	32	65	-36	-1
5	-20	28	-46	3	24	-30	24
6	-20	37	-28	12	-35	33	17
-5	-23	33	-30	17	-5	-4	20

$F(u,v)$

Example

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

Source image sample

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

DCT coefficients

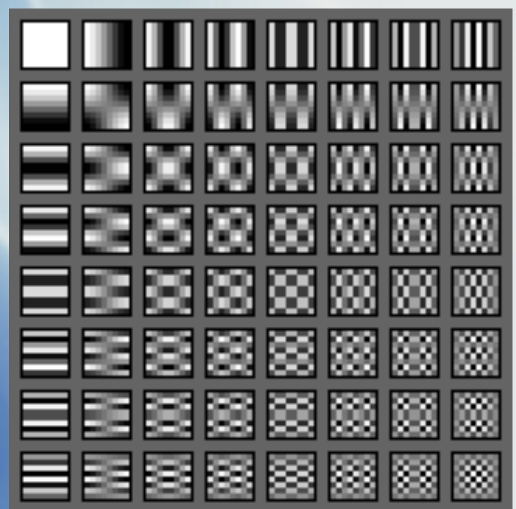
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Quantization table

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	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

Quantized coefficients

EXAMPLE OF 8X8 DCT BASIS FUNCTIONS



An 8 × 8 block from the Y image of 'Lena'

200	202	189	188	189	175	175	175
200	203	198	188	189	182	178	175
203	200	200	195	200	187	185	175
200	200	200	200	197	187	187	187
200	205	200	200	195	188	187	175
200	200	200	200	200	190	187	175
205	200	199	200	191	187	187	175
210	200	200	200	188	185	187	186

$f(i,j)$

515	65	-12	4	1	2	-8	5
-16	3	2	0	0	-11	-2	3
-12	6	11	-1	3	0	1	-2
-8	3	-4	2	-2	-3	-5	-2
0	-2	7	-5	4	0	-1	-4
0	-3	-1	0	4	1	-1	0
3	-2	-3	3	3	-1	-1	3
-2	5	-2	4	-2	2	-3	0

$F(u,v)$

32	6	-1	0	0	0	0	0
-1	0	0	0	0	0	0	0
-1	0	1	0	0	0	0	0
-1	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

$\hat{F}(u,v)$

512	66	-10	0	0	0	0	0
-12	0	0	0	0	0	0	0
-14	0	16	0	0	0	0	0
-14	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

$\tilde{F}(u,v)$

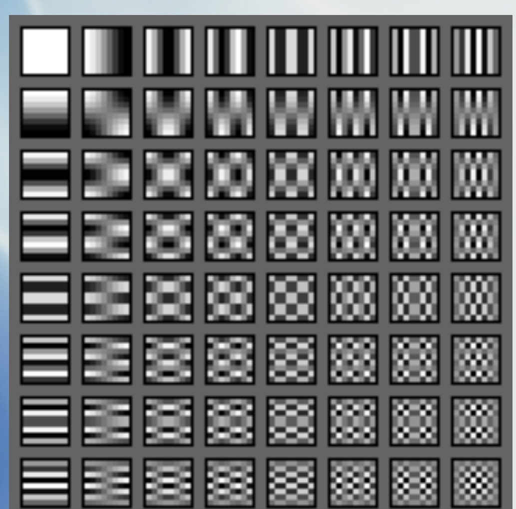
199	196	191	186	182	178	177	176
201	199	196	192	188	183	180	178
203	203	202	200	195	189	183	180
202	203	204	203	198	191	183	179
200	201	202	201	196	189	182	177
200	200	199	197	192	186	181	177
204	202	199	195	190	186	183	181
207	204	200	194	190	187	185	184

$\tilde{f}(i,j)$

1	6	-2	2	7	-3	-2	-1
-1	4	2	-4	1	-1	-2	-3
0	-3	-2	-5	5	-2	2	-5
-2	-3	-4	-3	-1	-4	4	8
0	4	-2	-1	-1	-1	5	-2
0	0	1	3	8	4	6	-2
1	-2	0	5	1	1	4	-6
3	-4	0	6	-2	-2	2	2

$\epsilon(i,j) = f(i,j) - \tilde{f}(i,j)$

EXAMPLE OF 8X8 DCT BASIS FUNCTIONS



Another 8 × 8 block from the Y image of 'Lena'

70	70	100	70	87	87	150	187
85	100	96	79	87	154	87	113
100	85	116	79	70	87	86	196
136	69	87	200	79	71	117	96
161	70	87	200	103	71	96	113
161	123	147	133	113	113	85	161
146	147	175	100	103	103	163	187
156	146	189	70	113	161	163	197

$f(i,j)$

-80	-40	89	-73	44	32	53	-3
-135	-59	-26	6	14	-3	-13	-28
47	-76	66	-3	-108	-78	33	59
-2	10	-18	0	33	11	-21	1
-1	-9	-22	8	32	65	-36	-1
5	-20	28	-46	3	24	-30	24
6	-20	37	-28	12	-35	33	17
-5	-23	33	-30	17	-5	-4	20

$F(u,v)$

-5	-4	9	-5	2	1	1	0
-11	-5	-2	0	1	0	0	-1
3	-6	4	0	-3	-1	0	1
0	1	-1	0	1	0	0	0
0	0	-1	0	0	1	0	0
0	-1	1	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\hat{F}(u,v)$

-80	-44	90	-80	48	40	51	0
-132	-60	-28	0	26	0	0	-55
42	-78	64	0	-120	-57	0	56
0	17	-22	0	51	0	0	0
0	0	-37	0	0	109	0	0
0	-35	55	-64	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\tilde{F}(u,v)$

70	60	106	94	62	103	146	176
85	101	85	75	102	127	93	144
98	99	92	102	74	98	89	167
132	53	111	180	55	70	106	145
173	57	114	207	111	89	84	90
164	123	131	135	133	92	85	162
141	159	169	73	106	101	149	224
150	141	195	79	107	147	210	153

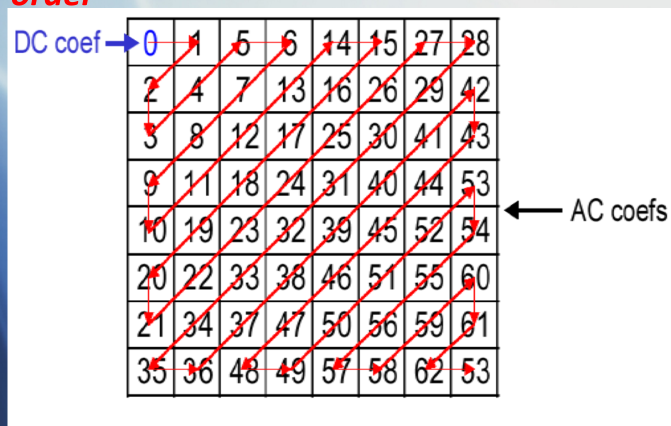
$\tilde{f}(i,j)$

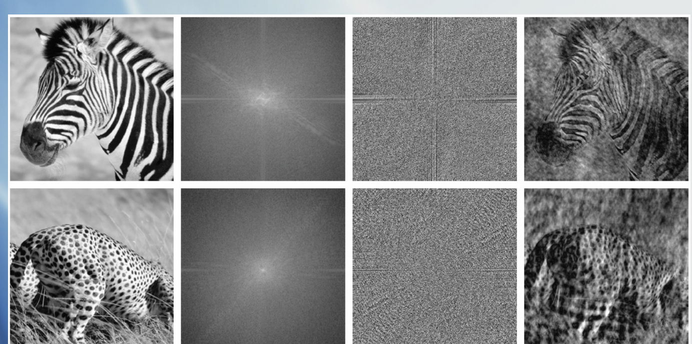
0	10	-6	-24	25	-16	4	11
0	-1	11	4	-15	27	-6	-31
2	-14	24	-23	-4	-11	-3	29
4	16	-24	20	24	1	11	-49
-12	13	-27	-7	-8	-18	12	23
-3	0	16	-2	-20	21	0	-1
5	-12	6	27	-3	2	14	-37
6	5	-6	-9	6	14	-47	44

$\epsilon(i,j) = f(i,j) - \tilde{f}(i,j)$

ENTROPY CODING OF DCT COEFS

The quantized DCT coefficients are first processed in *zigzag order*





- What happens if we swap the magnitude spectra ?
- Phase spectrum holds the spatial information (where things are),
- Phase spectrum is more important for perception than magnitude spectrum.