Lecture 7:
Mathematics of JPEG

Consider a $N \times N$ image f. Extend f to a $2M \times 2N$ image $\tilde{f}$, whose indices are taken from $[-M, M-1]$ and $[-N, N-1]$.

Define $f(k,l)$ for $-M \le k \le M-1$ and $-N \le l \le N-1$ such that

$$f(-k-1,-l-1) = f(k,l) \quad \} \text{ Reflection about } (-1/2,-1/2)$$

$$\left.\begin{array}{r} f(-k-1,l) = f(k,l) \\ f(k,l-1) = f(k,l) \end{array}\right\} \text{Reflection about the axis } k = -1/2 \text{ and } l = -1/2$$

Example:

$l = -1/2$

$$\begin{array}{cccc|ccc|l}
 & 9 & 8 & 7 & 7 & 8 & 9 & k=-3 \\
 & 6 & 5 & 4 & 4 & 5 & 6 & k=-2 \\
 & 3 & 2 & 1 & 1 & 2 & 3 & k=-1 \\
\hline
 & 3 & 2 & 1 & 1 & 2 & 3 & k=0 \\
 & 6 & 5 & 4 & 4 & 5 & 6 & k=1 \\
 & 9 & 8 & 7 & 7 & 8 & 9 & k=2 \\
 & l=-3 & l=-2 & l=-1 & l=0 & l=1 & l=2 &
\end{array}$$

$k = -1/2$

$f(-2,-2)$

$=$

$f(1,1)$

Reflection about $(-1/2,-1/2)$.

$f(-1,1)$

$=$

$f(0,1)$

Reflection about the axis $k = -1/2$.

Make the extension as a reflection about $(0,0)$, the axis $k=0$ and the axis $l=0$.

Done by shifting the image by $(1/2, 1/2)$

After shifting

$$
\begin{array}{cccccc|l}
9 & 8 & 7 & 7 & 8 & 9 & \frac{1}{2}+(-3) \\
6 & 5 & 4 & 4 & 5 & 6 & \frac{1}{2}+(-2) \\
3 & 2 & 1 & 1 & 2 & 3 & \frac{1}{2}+(-1) \\
3 & 2 & 1 & 1 & 2 & 3 & \frac{1}{2}+0 \\
6 & 5 & 4 & 4 & 5 & 6 & \frac{1}{2}+1 \\
9 & 8 & 7 & 7 & 8 & 9 & \frac{1}{2}+2 \\
\hline
\frac{1}{2}+-3 & \frac{1}{2}+-2 & \frac{1}{2}+-1 & \frac{1}{2}+0 & \frac{1}{2}+1 & \frac{1}{2}+2 &
\end{array}
$$

k

l

Now, we compute the DFT of (shifted) $\tilde{f}$:

$$F(m,n) = \frac{1}{(2M)(2N)} \sum_{k=-M}^{M-1} \sum_{l=-N}^{N-1} f(k,l) e^{-j\frac{2\pi}{2M}m(k+\frac{1}{2})} e^{-j\frac{2\pi}{2N}n(l+\frac{1}{2})}$$

$$= \frac{1}{4MN} \sum_{k=-M}^{M-1} \sum_{l=-N}^{N-1} f(k,l) e^{-j(\frac{\pi}{M}m(k+\frac{1}{2}) + \frac{\pi}{N}n(l+\frac{1}{2}))}$$

$$= \frac{1}{4MN} (\underbrace{\sum_{k=-M}^{-1} \sum_{l=-N}^{-1}}_{A_1} + \underbrace{\sum_{k=-M}^{-1} \sum_{l=0}^{N-1}}_{A_2} + \underbrace{\sum_{k=0}^{M-1} \sum_{l=-N}^{-1}}_{A_3} + \underbrace{\sum_{k=0}^{M-1} \sum_{l=0}^{N-1}}_{A_4})$$

$$f(k,l) e^{-j(\frac{\pi}{M}m(k+\frac{1}{2}) + \frac{\pi}{N}n(l+\frac{1}{2}))}$$

After some messy simplication, we can get:

$$A_1 + A_2 + A_3 + A_4 = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f(k,l) \cos\left[\frac{m\pi}{M}\left(k+\frac{1}{2}\right)\right] \cos\left[\frac{n\pi}{N}\left(l+\frac{1}{2}\right)\right]$$

Definition: (Even symmetric discrete cosine transform [EDCT])

Let f be a $M \times N$ image, whose indices are taken as $0 \leq k \leq M-1$ and $0 \leq l \leq N-1$. The **even symmetric discrete cosine transform (EDCT)** of f is given by:

$$\hat{f}_{ec}(m,n) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f(k,l) \cos\left[\frac{m\pi}{M}\left(k+\frac{1}{2}\right)\right] \cos\left[\frac{n\pi}{N}\left(l+\frac{1}{2}\right)\right]$$

with $0 \leq m \leq M-1, 0 \leq n \leq N-1$

Remark:
- Smart idea to get a decomposition consisting only of cosine function (by reflection and shifting!)
- Can be formulated in matrix form
- Again, it is a separable image transformation.

- The inverse of EDCT can be explicitly computed. More specifically, the **inverse EDCT** is defined as:

$$f(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} C(m)C(n)\hat{f}_{ec}(m,n)\cos\frac{\pi m(2k+1)}{2M}\cos\frac{\pi n(2l+1)}{2N} \qquad (**)$$

where $C(0) = 1, C(m) = C(n) = 2$ for $m, n \neq 0$

Also involving cosine functions only!

- Formula (**) can be expressed as matrix multiplication:

$$f = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \hat{f}_{ec}(m,n)\vec{T}_m \vec{T'}_n^T$$

elementary images under EDCT!

where: $\vec{T}_m = \begin{pmatrix} T_m(0) \\ T_m(1) \\ \vdots \\ T_m(M-1) \end{pmatrix}, \vec{T'}_n = \begin{pmatrix} T'_n(0) \\ T'_n(1) \\ \vdots \\ T'_n(N-1) \end{pmatrix}$ with $T_m(k) = C(m)\cos\frac{\pi m(2k+1)}{2M}$

and $T'_n(k) = C(n)\cos\frac{\pi n(2k+1)}{2N}$.

This is what JPEG does!!

Something similar can be developed:

Definition: (Odd symmetric discrete cosine transform [ODCT])

Let f be a $M \times N$ image, whose indices are taken as $0 \le k \le M-1$ and $0 \le l \le N-1$. The **odd symmetric discrete cosine transform (ODCT)** of f is given by:

$$\hat{f}_{oc}(m,n) = \frac{1}{(2M-1)(2N-1)} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} C(k)C(l)f(k,l) \cos\frac{2\pi mk}{2M-1} \cos\frac{2\pi nl}{2N-1}$$

where $C(0) = 1$ and $C(k) = C(l) = 2$ for $k, l \neq 0$, $0 \le m \le M-1, 0 \le n \le N-1$.

The **inverse ODCT** is given by:

$$f(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} C(m)C(n)\hat{f}_{oc}(m,n) \cos\frac{2\pi mk}{2M-1} \cos\frac{2\pi nl}{2N-1}$$

where $C(0) = 1, C(m) = C(n) = 2$ if $m, n \neq 0$

Understanding convolution:

Recall: Discrete convolution:

$$\underset{g * I(n,m)}{\overset{\shortparallel}{v(n,m)}} = \underbrace{\sum_{n'=0}^{N-1} \sum_{m'=0}^{N-1} g(n-n', m-m')\, I(n', m')}_{\text{Linear combination of pixel values of } I}$$

In particular, if $g(k,l)$ is only non-zero around $(0,0)$, then, $g * I(n,m)$ is a linear combination of pixel value of I around (n,m)!!

Image enhancement in the frequency domain:

Goal: 1. Remove $\underbrace{\text{high-frequency components}}_{\text{noise}}$ (low-pass filter) for image denoising.

2. Remove $\underbrace{\text{low-frequency components}}_{\text{edges}}$ (high-pass filter) for the extraction of image details.

High/Low frequency components of $\hat{F}$

Let F be a $N \times N$ image, N = even. Let $\hat{F}$ = DFT of F.

$$\therefore \hat{F}(k, l) = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} F(m,n)\, e^{-j 2\pi \cdot \left(\frac{mk + nl}{N}\right)}$$

Fourier coefficients of F at (k, l)

Observe that : for $0 \le k, l \le \frac{N}{2} - 1$

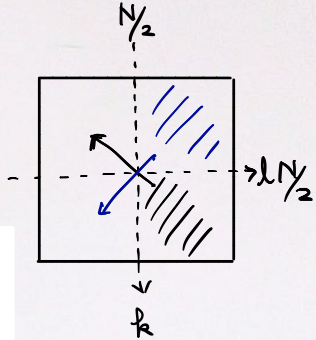
$$\hat{F}\left(\frac{N}{2} + k, \frac{N}{2} + l\right) = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} F(m,n)\, e^{-j \frac{2\pi}{N}\left(m\left(\frac{N}{2} + k\right) + n\left(\frac{N}{2} + l\right)\right)}$$

$$= \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} F(m,n) (-1)^{m+n} e^{-j \frac{2\pi}{N}(m(-k) + n(-l))}$$

$$= \overline{\frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} F(m,n)\, e^{-j\frac{2\pi}{N}\left(m\left(\frac{N}{2}-k\right)+n\left(\frac{N}{2}-l\right)\right)}}$$

$$= \overline{\hat{F}\left(\frac{N}{2}-k,\ \frac{N}{2}-l\right)}$$

$\therefore$ Computing part of $\hat{F}$ can determine the rest !!



We have :

$$\begin{aligned}
F(m,n) = &\sum_{0\le k,l\le \frac{N}{2}-1}\left[\hat{F}\left(\frac{N}{2}+k,\frac{N}{2}+l\right)e^{j\frac{2\pi}{N}[(\underline{\frac{N}{2}+k})m+(\underline{\frac{N}{2}+l})n]}\right]\\
&+\sum_{1\le k,l\le \frac{N}{2}-1}\left[\overline{\hat{F}\left(\frac{N}{2}+k,\frac{N}{2}+l\right)}e^{j\frac{2\pi}{N}[(\underline{\frac{N}{2}-k})m+(\underline{\frac{N}{2}-l})n]}\right]\\
&+\sum_{0\le k,l\le \frac{N}{2}-1}\left[\hat{F}\left(\frac{N}{2}+k,\frac{N}{2}-l\right)e^{j\frac{2\pi}{N}[(\underline{\frac{N}{2}+k})m+(\underline{\frac{N}{2}-l})n]}\right]\\
&+\sum_{1\le k,l\le \frac{N}{2}-1}\left[\overline{\hat{F}\left(\frac{N}{2}+k,\frac{N}{2}-l\right)}e^{j\frac{2\pi}{N}[(\underline{\frac{N}{2}-k})m+(\underline{\frac{N}{2}+l})n]}\right]\\
&+\sum_{0\le l\le \frac{N}{2}-1}\hat{F}\left(0,\frac{N}{2}+l\right)e^{j\frac{2\pi}{N}[(\frac{N}{2}+l)n]}+\sum_{1\le l\le \frac{N}{2}-1}\overline{\hat{F}\left(0,\frac{N}{2}+l\right)}e^{j\frac{2\pi}{N}[(\frac{N}{2}-l)n]}\\
&+\sum_{0\le k\le \frac{N}{2}-1}\hat{F}\left(\frac{N}{2}+k,0\right)e^{j\frac{2\pi}{N}[(\frac{N}{2}+k)m]}+\sum_{1\le k\le \frac{N}{2}-1}\overline{\hat{F}\left(\frac{N}{2}+k,0\right)}e^{j\frac{2\pi}{N}[(\frac{N}{2}-k)m]}+\hat{F}(0,0)
\end{aligned}$$

Observation:

1. When k and l are close to $N/2$, $\hat{F}\left(\frac{N}{2}+k, \frac{N}{2}+l\right)$ is associated to $e^{-j\frac{2\pi}{N}\left(\left(\frac{N}{2}+k\right)m+\left(\frac{N}{2}+l\right)n\right)}$

   $\frac{N}{2}+k \approx N$, $\frac{N}{2}+l \approx N$

   $\approx e^{-j\frac{2\pi}{N}(k'm+l'n)}$ where $(k',l') \approx (0,0)$

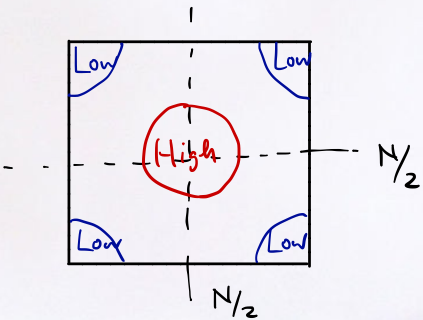
   $= \cos\left(\frac{2\pi}{N}(k'm+l'n)\right) + i\sin\left(\frac{2\pi}{N}(k'm+l'n)\right)$

   Low-frequency if $(k,l) \approx (0,0)$

   ∴ Fourier coefficients at the bottom right are associated to low frequency components!

2. Similarly, we can check that Fourier coefficients at the 4 corners are associated to low frequency components.

3. Fourier coefficients in the middle are associated to high-frequency components



!!

∴ High-pass filtering
$=$ Remove coefficients at 4 corners

Low-pass filtering
$=$ Remove coefficients at the center

Centralisation:

Assume periodic conditions on F.

We can let $\tilde{F}(u,v) = \hat{F}(u - \frac{N}{2}, v - \frac{N}{2})$ where $0 \leq u \leq N-1$

$0 \leq v \leq N-1$

Then, High-frequency components are located at 4 corners of $\tilde{F}(u,v)$

Low-frequency components are located at center of $\tilde{F}(u,v)$

Mathematically,

Consider the discrete Fourier transform of $(-1)^{x+y}F(x,y)$:

$$DFT(F(x,y)(-1)^{x+y})(u,v)$$

$$= \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{j\pi(x+y)} exp\left(-j2\pi\left(\frac{ux}{N} + \frac{vy}{N}\right)\right)$$

$$e^{j2\pi\frac{(\frac{N}{2}x + \frac{N}{2}y)}{N}}$$

$$= \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y) exp\left(-j2\pi\left(\frac{(u-N/2)x}{\cancel{M}\, N} + \frac{(v-N/2)y}{N}\right)\right)$$

$$= \hat{F}\left(u - \frac{N}{2}, v - \frac{N}{2}\right) = \tilde{F}(u. \quad , v \quad .)$$

Therefore, to compute $\tilde{F}(u,v)$, we can compute DFT of $(-1)^{x+y}f(x,y)$.

Definition 3.5: A **low-pass filter (LPF)** (LPF) leaves low frequencies unchanged, while attenuating the high frequencies.

A **high-pass filter (HPF)** leaves high frequencies unchanged, while attenuating the low frequencies.

Basic steps of filtering in the frequency domain

1. Multiply $f(x, y)$ by $(-1)^{x+y}$.
2. Compute $\tilde{F}(u, v) = DFT(f(x, y)(-1)^{x+y})(u, v)$.
3. Multiply $\tilde{F}$ by a real "filter" function $H(u, v)$ to get

   $$G(u, v) = H(u, v)\tilde{F}(u, v)$$

   (point-wise multiplication, but not matrix multiplication)
4. Compute inverse DFT of $G(u, v)$.
5. Take real part of the result in Step 4.
6. Multiply the result in Step 5 by $(-1)^{x+y}$.

Remark: 1. H is taken either to remove high-frequency coefficients / low-frequency coefficients.

2. $\mathcal{F}^{-1}(G(u,v)) = g(x,y) = N^2 \mathcal{F}^{-1}(H(u,v)) * \mathcal{F}^{-1}(\tilde{F}(u,v)) = N^2\, h * \tilde{f}(x,y)$

Filtering in the frequency domain = Linear filtering in the spatial domain!

Example of Low-pass filters for image denoising

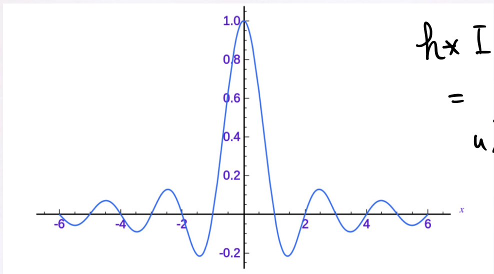
Assume that we work on the centered spectrum!

That is, consider $\hat{F}(u,v)$ where $-N/2 \leq u \leq N/2 - 1$, $-N/2 \leq v \leq N/2 - 1$.

1 Ideal low pass filter (ILPF):

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) := u^2+v^2 \leq D_0^2 \\ 0 & \text{if } D(u,v) > D_0^2 \end{cases}$$

In 1-dim cross-section, $\mathcal{F}^{-1}(H(u,v))$ looks like:



$$h * I(x,y) = \sum_{u,v} h(x-u, y-v)\, I(u,v)$$

every pixel values of I has an effect on $h * I(x,y)$!!

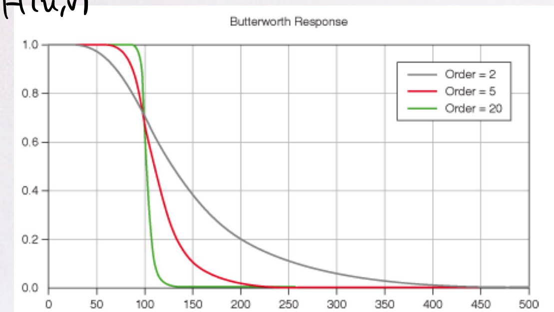
Good: Simple

Bad : Produce ringing effect!

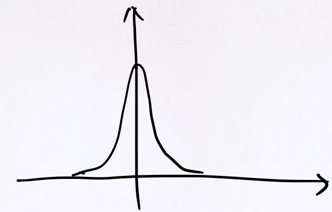
2. Butterworth low-pass filter (BLPF) of order n ($n \geq 1$ integer):

$$H(u,v) = \frac{1}{1 + \left(D(u,v)/D_0\right)^n}$$

$H(u,v)$ in 1-dim



$\mathcal{F}^{-1}(H(u,v))$ in 1-dim

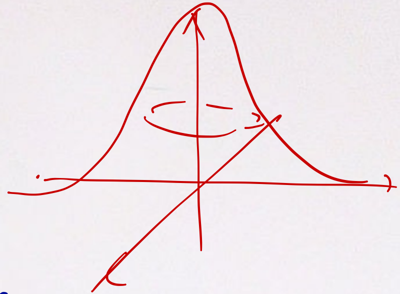


Good: Produce less / no visible ringing effect if n is carefully chosen!!

3. Gaussian low-pass filter

$$H(u,v) = \exp\left(-\frac{\overset{u^2+v^2}{D(u,v)}}{2\sigma^2}\right)$$

σ = spread of the Gaussian function



F.T. of Gaussian is also Gaussian!!

Good: No visible ringing effect!!