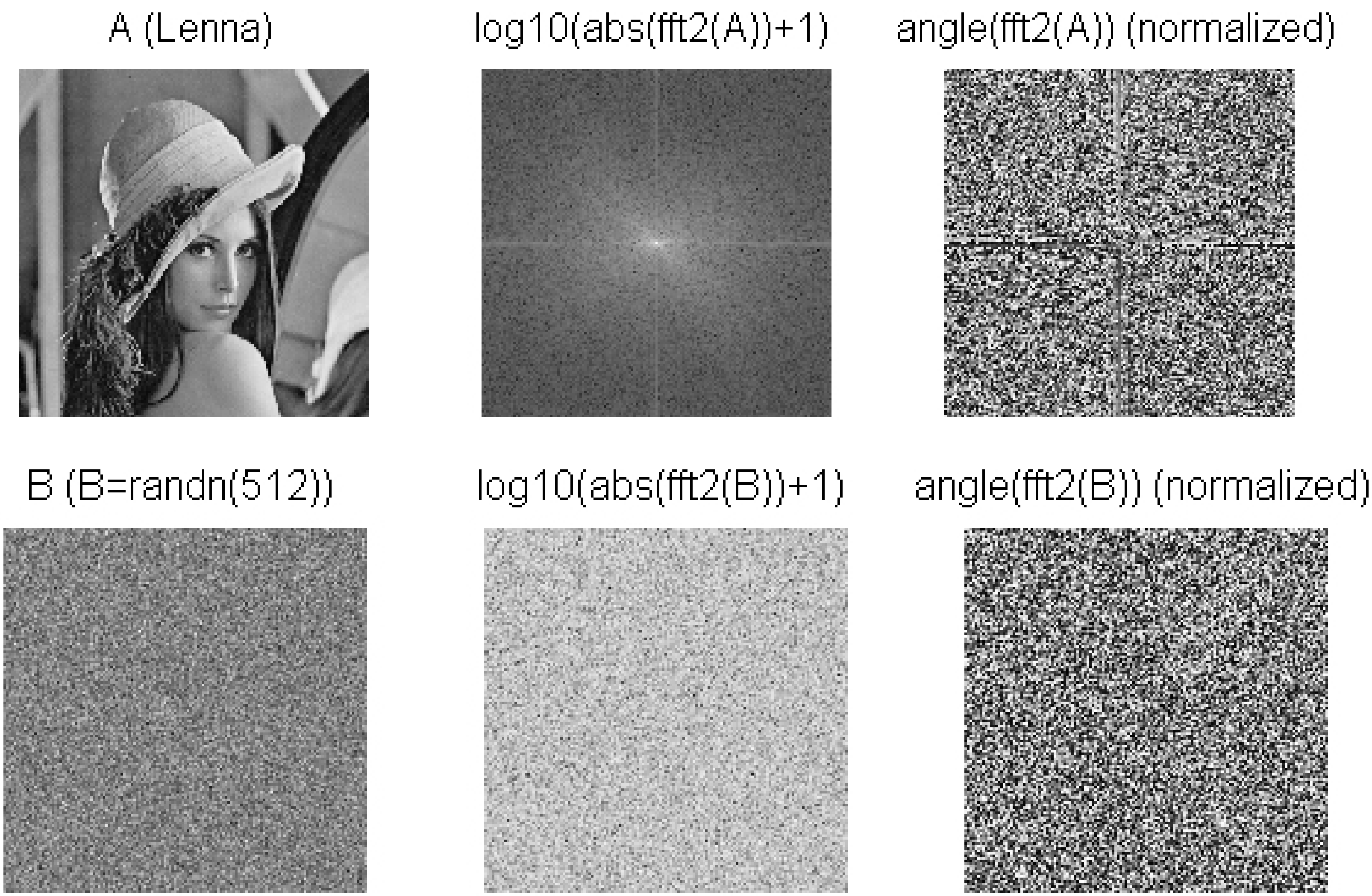
Summary of Lecture 7

- In lecture 7 we learnt the 2-D DFT of two dimensional finite extent sequences.
- We learnt how to calculate convolutions using DFTs.
- We learnt about basic properties of the DFTs of natural images.

2-D DFT and Convolution

- The DFT can be computed with a fast algorithm and it is sometimes beneficial to do the convolution of two sequences $\mathbf{A}$ $(M_1 \times N_1)$ and $\mathbf{B}$ $(M_2 \times N_2)$ via $[M_1 + M_2 + 1, N_1 + N_2 + 1]$ point DFTs.
- Speed improvements are only possible if $both$ sequences have large dimensions. Otherwise convolutions are better implemented via the convolution sum.

DFTs of Natural Images

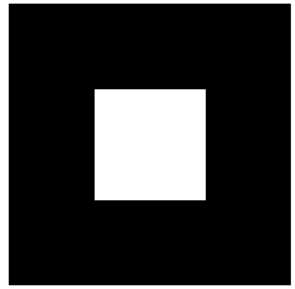


2-D Low-Pass Filtering of Images

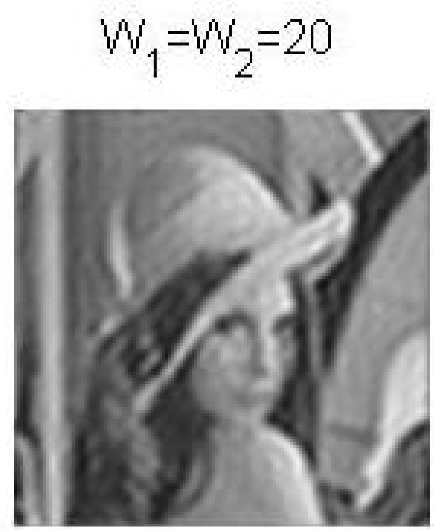
We will be interested in two ways of implementing low-pass filtering for images:

- By defining "windows" in the DFT domain, selecting low frequency DFT coefficients of images and inverse transforming.
- By defining low-pass filters in spatial domain and obtaining filtered images by the convolution sum.

Low-Pass Filtering by DFT windows



w for $W_1 = W_2 = 100$ (normalized and fftshifted)



Low-Pass Filtering in Spatial Domain

Low pass filtering operations in spatial domain can be thought of as local averaging operations. Let

$$L(m,n) = \begin{cases} \frac{1}{(2W+1)^2} & -W \leq m,n \leq W \\ 0 & \text{otherwise} \end{cases}$$

Consider $\mathbf{C} = \mathbf{L} \otimes \mathbf{A}$ where $\mathbf{A}$ is an image.

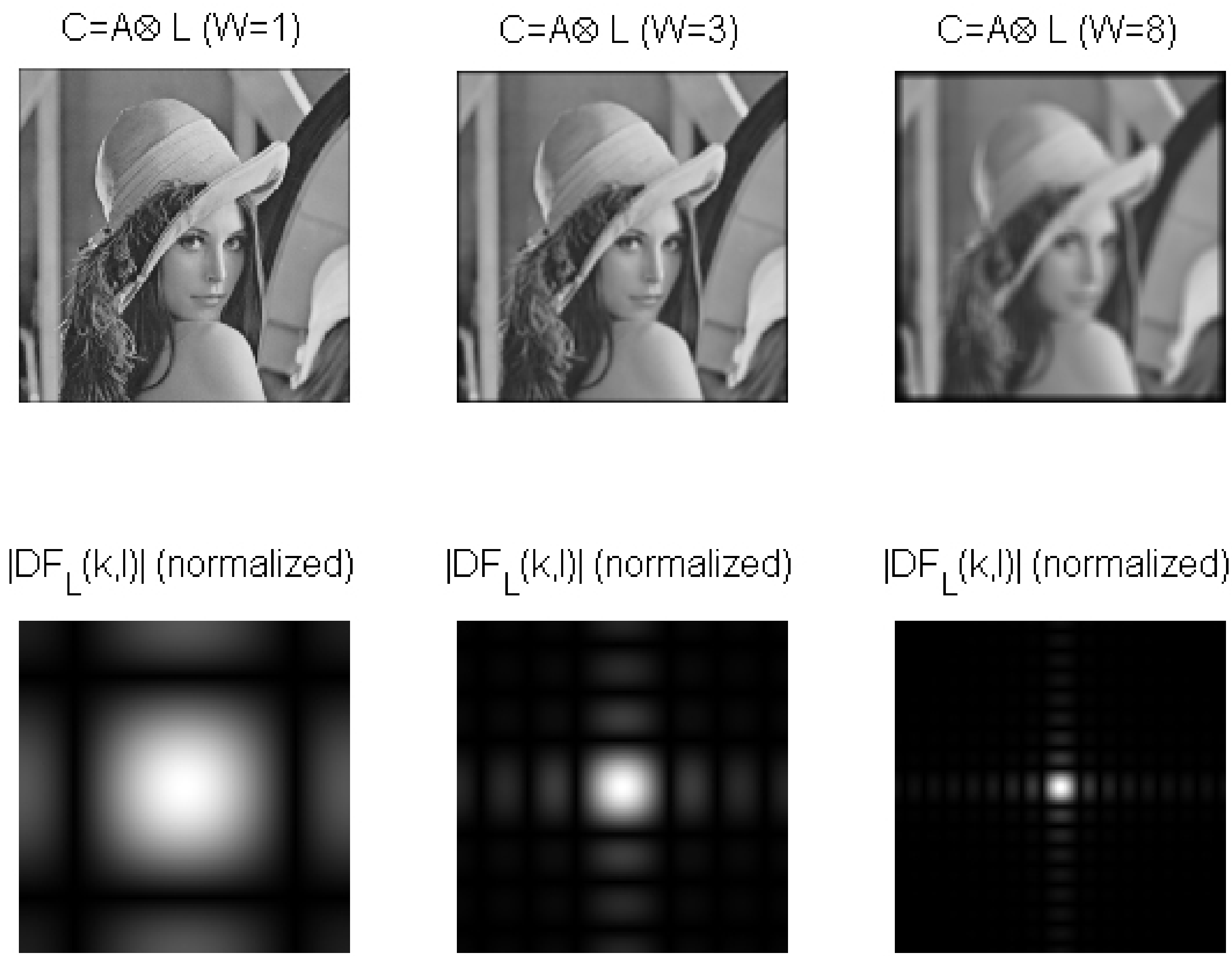
$$\begin{aligned} C(m,n) &= \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} A(m-k,n-l)L(k,l) \\ &= \frac{1}{(2W+1)^2} \sum_{k=-W}^{W} \sum_{l=-W}^{W} A(m-k,n-l) \end{aligned} \tag{1}$$

This is a local average if W is much smaller than the dimensions of $\mathbf{A}$. For $W=1$ Equation 1 becomes:

$$\begin{aligned} C(m,n) &= \frac{1}{9}(A(m-1,n-1) + A(m-1,n) + A(m-1,n+1) \\ &+ A(m,n-1) + A(m,n) + A(m,n+1) \\ &+ A(m+1,n-1) + A(m+1,n) + A(m+1,n+1)) \end{aligned}$$

Note that $\mathbf{L}$ becomes more and more low-pass as we increase W.

Example



C=A⊗ L (W=1)
C=A⊗ L (W=3)
C=A⊗ L (W=8)
|DF_L(k,l)| (normalized)
|DF_L(k,l)| (normalized)
|DF_L(k,l)| (normalized)

Low-Pass Filtering in Spatial Domain

- Given the size $2W+1$ of the filter $\mathbf{L}$, one can design low-pass filters that implement more complicated forms of averaging using signal processing and statistical signal processing concepts.
- In this class, we will mainly concentrate on simple filters and not go through detailed filter design techniques.

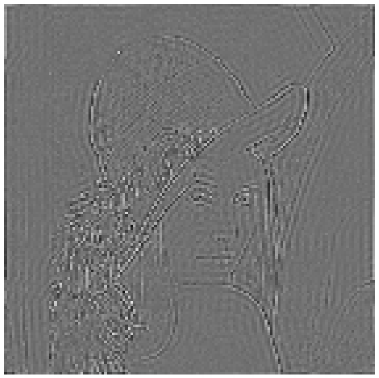
2-D High-Pass Filtering of Images

The high-pass filtered image can be thought of as the original image minus the low pass filtered image.

- High-pass filtering by DFT windows:
  - If $w(k,l)$ $(W_1 \times W_2)$ is a low-pass DFT window, simply define a high-pass window $h(k,l)$ by $h(k,l) = 1 - w(k,l)$.
- High-pass filtering in spatial domain:
  - If $\mathbf{L}$ is a low-pass filter of size W, simply define a high-pass filter $\mathbf{H}$ via $H(m,n) = \delta(m,n) - L(m,n)$.

High-Pass Filtering by DFT windows

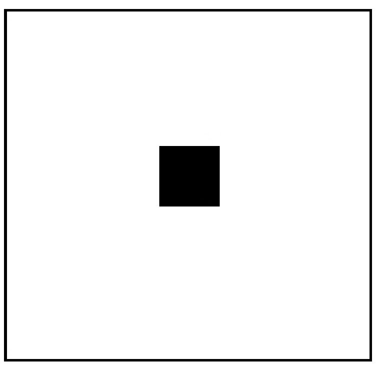
$W_1=W_2=40$ (nrml)



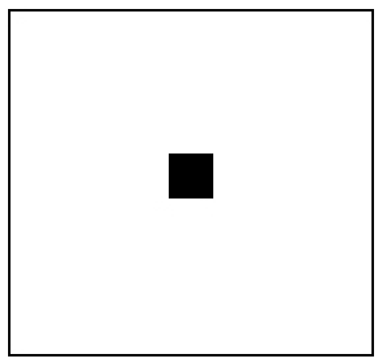
$W_1=W_2=30$ (nrml)

$W_1=W_2=20$ (nrml)

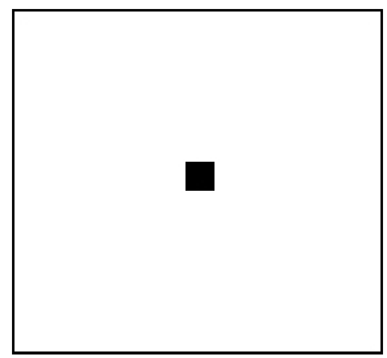
h (normalized)



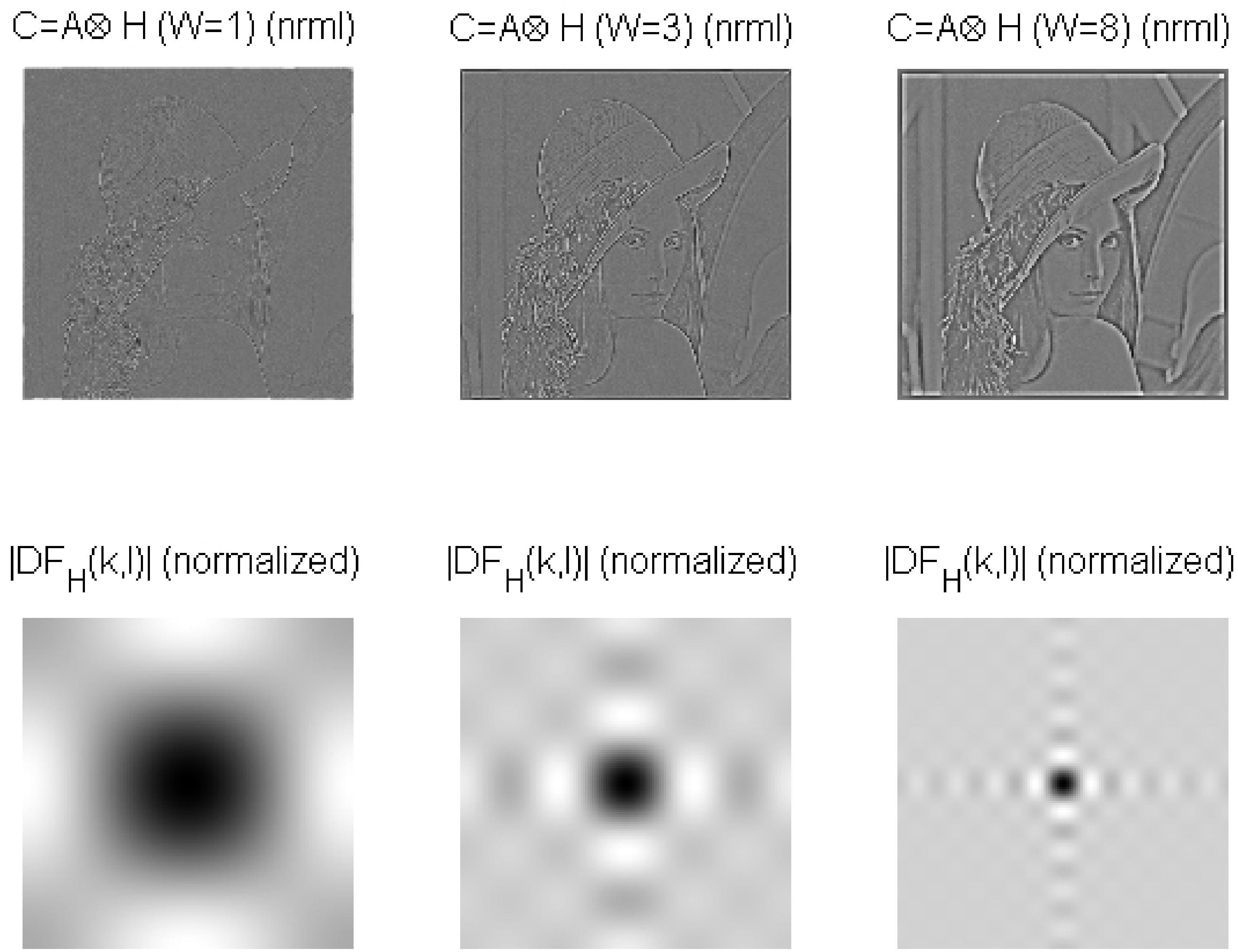
h (normalized)



h (normalized)



Spatial High-Pass Filtering



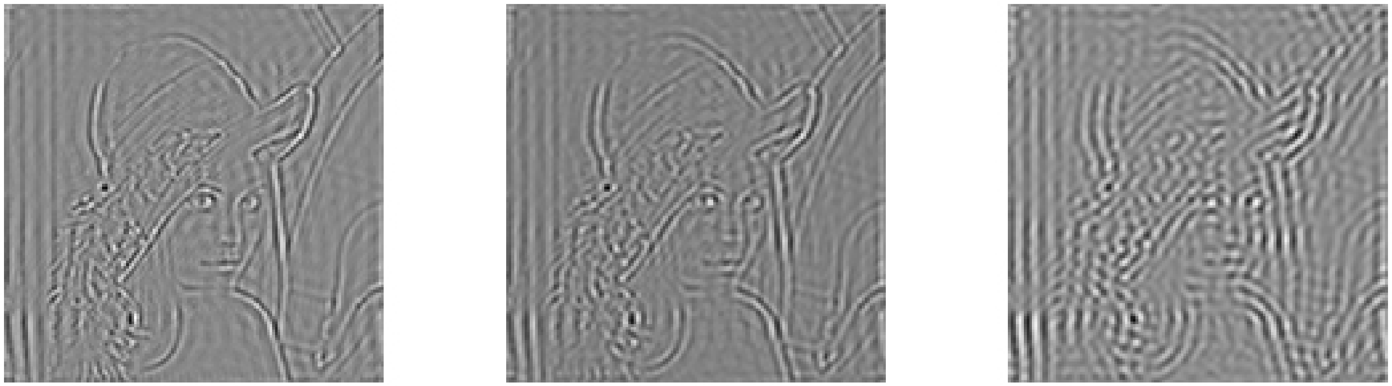
2-D Band-Pass Filtering of Images

The band-pass filtered image can be thought of as one low-pass filtered image minus another low pass filtered image:

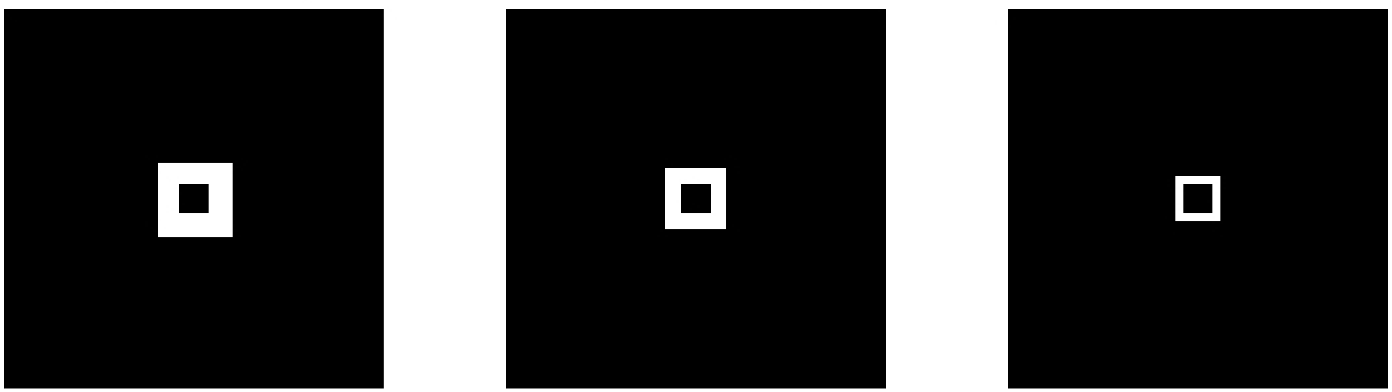
- Band-pass filtering by DFT windows:
  - If $w_1(k,l)$ $(W_1 \times W_2)$ and $w_2(k,l)$ $(W_1 + O1 \times W_2 + O2)$ are low-pass DFT windows, simply define a band-pass window $b(k,l)$ by $b(k,l) = w_2(k,l) - w_1(k,l)$.
- Band-pass filtering in spatial domain:
  - If $\mathbf{L}_1$ (size W) and $\mathbf{L}_2$ (size $W + O$) are low-pass filters with $\mathbf{L}_2$ being "lower-pass", simply define a band-pass filter $\mathbf{B}$ via $B(m,n) = L_1(m,n) - L_2(m,n)$.

Band-Pass Filtering by DFT windows

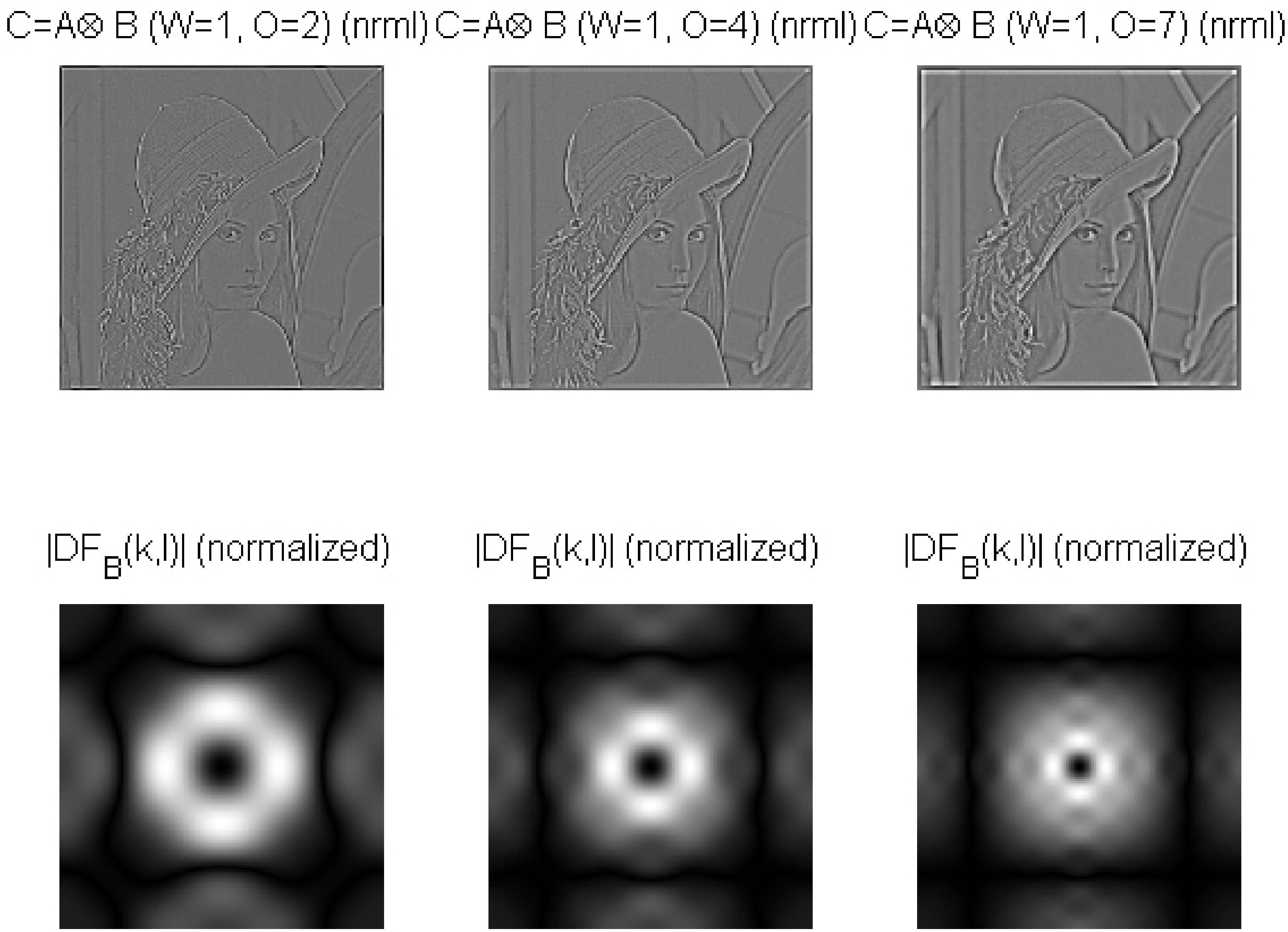
W1=W2=20, O1=O2=30 W1=W2=20, O1=O2=20 W1=W2=20, O1=O2=10



b (normalized) **b** (normalized) **b** (normalized)



Spatial Band-Pass Filtering



Filtering Convention

- Note that when $\mathbf{A}$ is $(M_1 \times N_1)$, $\mathbf{C} = \mathbf{L} \otimes \mathbf{A}$ is $(M_1 + W - 1 \times N_1 + W - 1)$.
- In general, we would like to keep $\mathbf{C}$ the same size as $\mathbf{A}$.
- Thus we crop a suitable portion of $\mathbf{C}$ and consider that as the low-pass filtered image.
- For the filters we have discussed, a good cropping region is $m = 0, \ldots, M_1 - 1$, etc.

Sampling and Antialiasing Filters

Barbara (512x512)

Lenna (512x512)

Sampling Without Antialiasing Filters

Barbara S1=S2=4 (128x128)

Lenna S1=S2=4 (128x128)

Barbara S1=S2=8 (64x64)

Lenna S1=S2=8 (64x64)

Sampling With Antialiasing Filters

Barbara S1=S2=4 (W1=W2=63)

Lenna S1=S2=4 (W1=W2=63)

Barbara S1=S2=8 (W1=W2=31)

Lenna S1=S2=8 (W1=W2=31)

Noise Removal

Consider the scenario where an image $\mathbf{A}$ is *corrupted* with additive noise to yield an image $\mathbf{B}$:

$$\mathbf{B} = \mathbf{A} + \mathbf{N} \tag{2}$$

A+10*randn(512) (nrml)

A+20*randn(512) (nrml)

A+30*randn(512) (nrml)

Noise Removal in DFT domain

We already know that natural images have dominant low frequency DFT coefficients. Intuitively, we can make the following observations.

- Assuming noise is not "accessive" at low frequencies we expect:

$$\begin{aligned} DF_B(k,l) &= DF_A(k,l) + DF_N(k,l) \\ DF_B(k,l) &\tilde{=} DF_A(k,l) \end{aligned} \tag{3}$$

  since $|DF_A(k,l)|$ is large at low frequencies.

- At high frequencies we expect:

$$\begin{aligned} DF_B(k,l) &= DF_A(k,l) + DF_N(k,l) \\ DF_B(k,l) &\tilde{=} DF_N(k,l) \end{aligned} \tag{4}$$

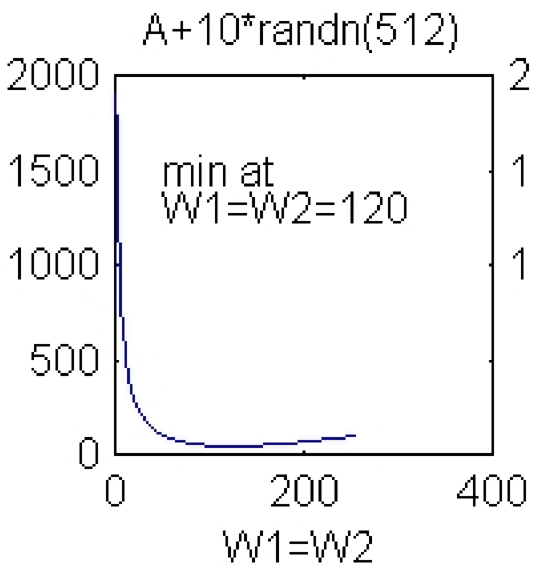
  since $|DF_A(k,l)|$ is small at high frequencies.

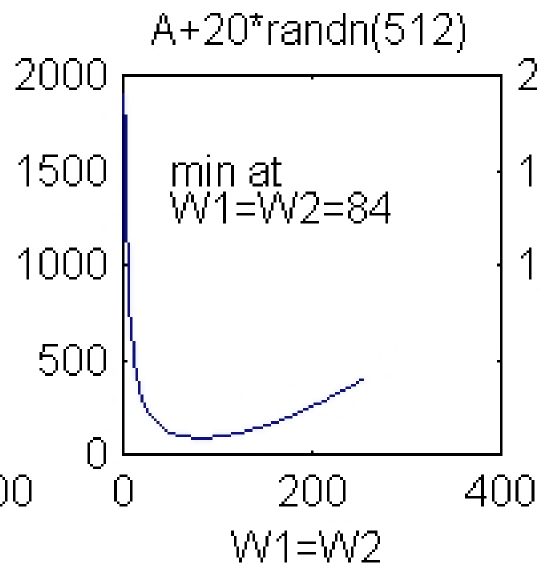
- We can reduce the amount of noise in $\mathbf{B}$ by low-pass filtering.

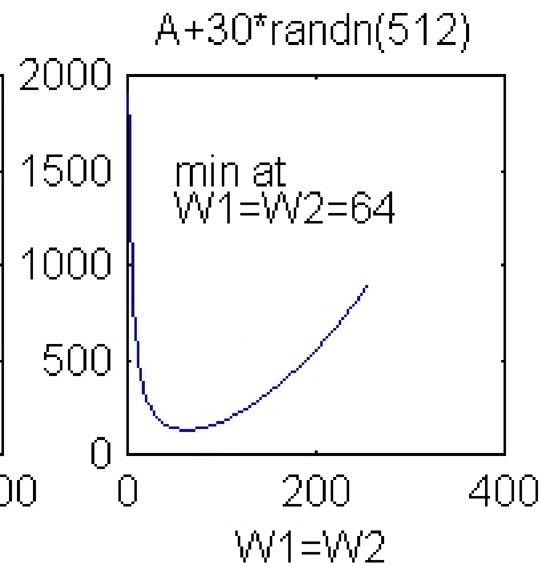
Noise Removal By Low-Pass Filtering

- Given noisy image, low-pass filter it to obtain $\mathbf{C} = \mathbf{B} \otimes \mathbf{L}$ or $DF_C(k,l) = DF_B(k,l)w(k,l)$, where $\mathbf{L}$ is a low-pass filter and $w(k,l)$ is a low-pass DFT window.
- In general, determining the parameters of the filters is difficult and is done by trial/error (say by judging the visual quality of $\mathbf{C}$) or based on certain assumptions/models.
- For illustration purposes we will determine the *best* parameters for our filters based on the mean squared error between $\mathbf{C}$ and $\mathbf{A}$. Note that this is not possible in practice as access to the original image is not possible.

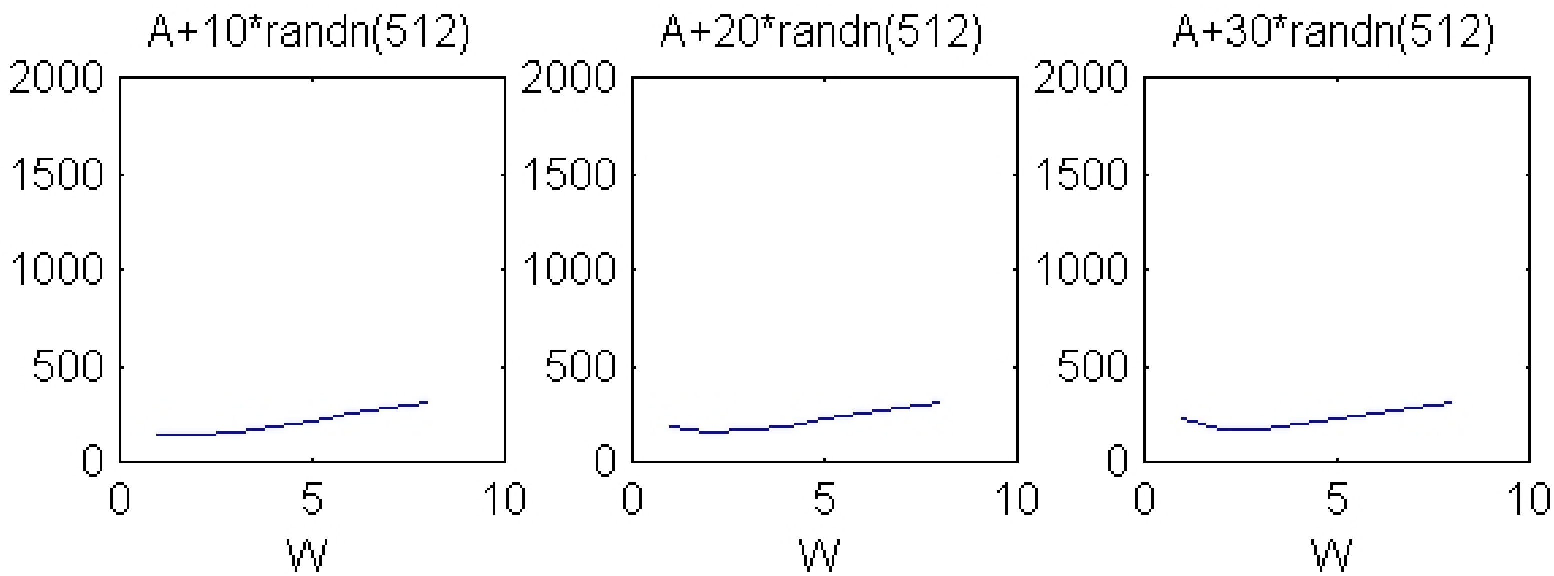
Noise Removal by DFT windows







Noise Removal by Spatial Low-Pass Filters



Summary

- In this lecture we learnt how to low-pass, high-pass and band-pass filter images in two different ways.
- We considered two filtering applications:
  - Subsampling by low-pass antialiasing filters.
  - Noise reduction/removal by low-pass filters

Homework VIII

1. Low-pass, band-pass and high-pass filter your image both spatially and with DFT windows. Use at least three different parameters for low, band and high pass filtering. Present your results as they have been presented in this lecture (see for e.g. pages 10 and 11).
2. Subsample your image by 4 and 8 in each direction, with and without antialiasing using low-pass DFT windows. Make sure you pick the correct parameters for the windows. (Hint: your images are not square). Show the parameters used as well as the resulting images. Comment on the results.
3. Do the noise reduction processing I did on pages 22 and 23. Start by adding noise to your image etc. (See above hint for low-pass DFT window parameters.) Present your results as they have been presented in this lecture.

References

[1] A. K. Jain, *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice Hall, 1989.