2D Discrete Fourier Transform

In these lecture notes the figures have been removed for copyright reasons. References to figures are given instead, please check the figures yourself as given in the course book, 3rd edition.

RRY025: Image processing

Eskil Varenius

Monday: Plan

- Brief repetition: What is 1D continuous FT
- The 2D Discrete Fourier Transform
- Important things in a discrete world:
  - Freq. Smoothing and leakage
  - Aliasing
  - Centering
  - Edge effects
  - Convolution
- Two hours of Matlab exercises

Repetition: The 1D continuous FT

See your handwritten notes.
Also see Fig. 4.1 in Book

… what if we have discrete 2D signals (images)?

The 2D Discrete Fourier Transform

Defined for a sampled image *f(x, y)* of MxN pixels:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)}$$ (Book: eq. 4.5-15)

where x = 0, 1, 2…M-1, y = 0,1,2…N-1 and u = 0, 1, 2…M-1, v = 0, 1, 2…N-1.

How do you get back? Use the **Inverse transform**!

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(ux/M + vy/N)}$$ (Book: eq. 4.5-16)

Some differences to continuous FT

- DFT works on **finite** images with MxN pixels
  → Frequency smoothing, freq. leakage

- DFT uses discrete **sampled** images i.e. pixels
  → Aliasing

- DFT assumes **periodic** boundary conditions
  → Centering, Edge effects, Convolution

Frequency smoothing and leakage

Images have borders, they are truncated (**finite**).

This causes freq. smoothing and freq. leakage.

Aliasing

Images consists of pixels, they are **sampled**.

Too few pixels → fake signals (aliasing)!

How do you avoid aliasing?

Aliasing in 1D

Fig 4.10

Aliasing in 2D

Fig 4.16

Aliasing explained with DFT (1D)

Fig 4.6

Aliasing explained with DFT (1D)

Fig 4.7

Aliasing explained with DFT (1D)

Fig 4.8

Aliasing explained with DFT (1D)

Fig 4.9

Aliasing explained with DFT - in 2D!

Fig 4.15

Aliasing: Take home message

How do you avoid aliasing?
1. Make sure signal is band limited. How?
2. Then: sample with enough pixels!

What is enough pixels?
Nyquist Theorem: The signal must be measured at least twice per period, i.e.:

$$\Delta x < \frac{1}{2u_{\max}} \quad \text{and} \quad \Delta y < \frac{1}{2v_{\max}}$$

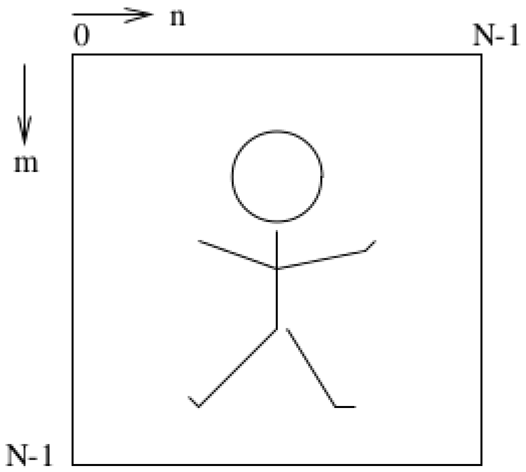
2 min pause to discuss

- What is freq. smoothing and freq. leakage?
  Why is it important?

- What is aliasing?
  Why is it important?
  How do you avoid aliasing?

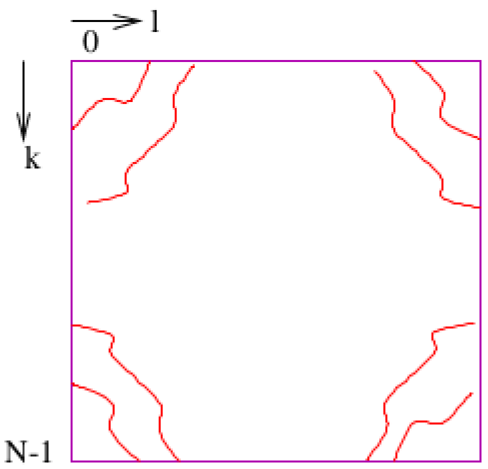
Centering: Looking at DFTs

Object
f(m,n)

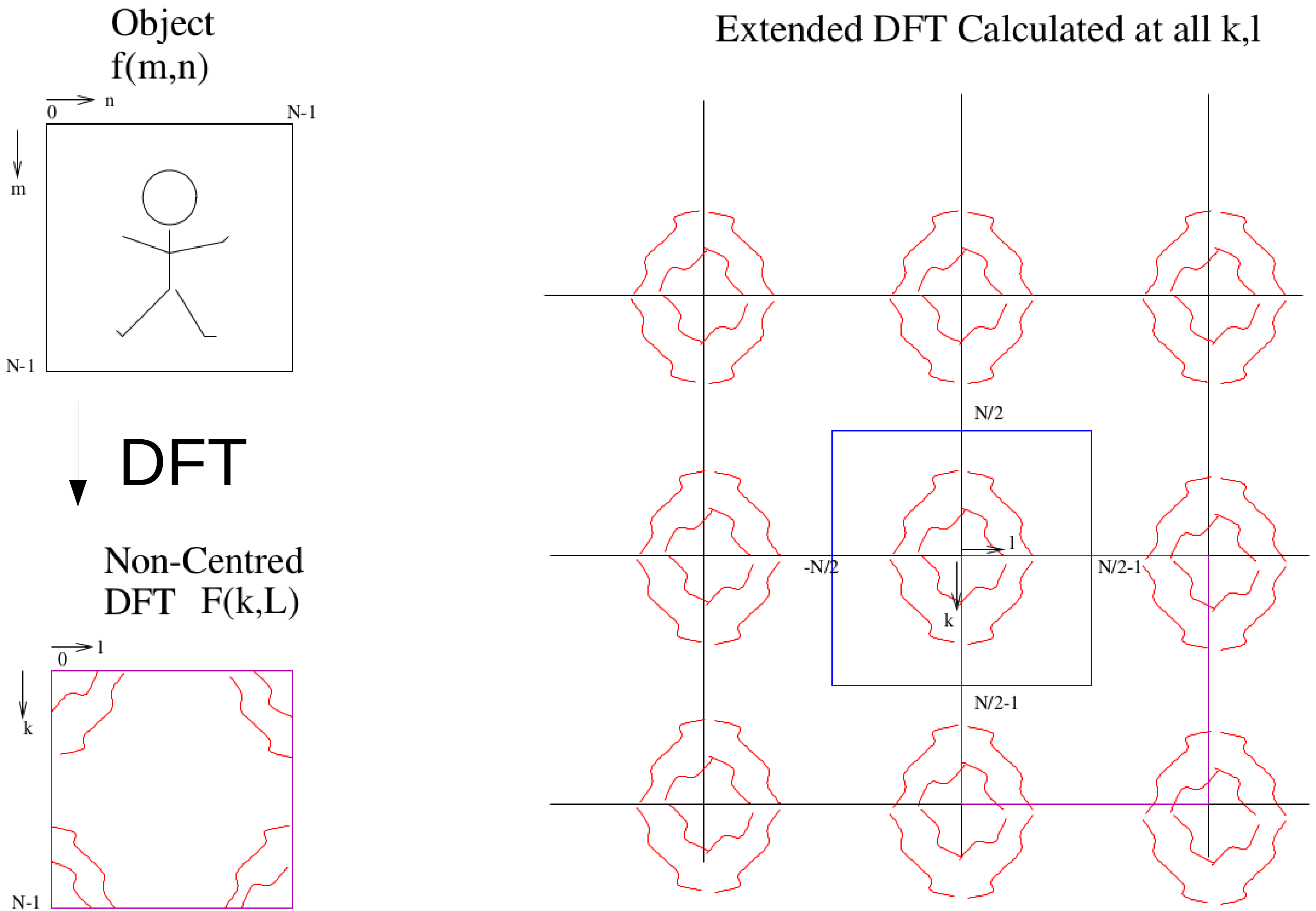


DFT

Non-Centred
DFT F(k,L)



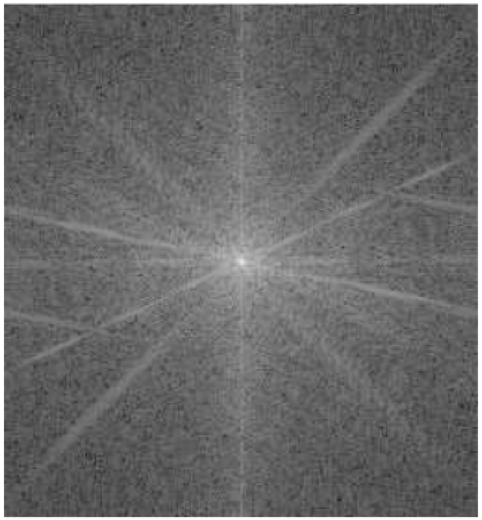
Centering: Looking at DFTs



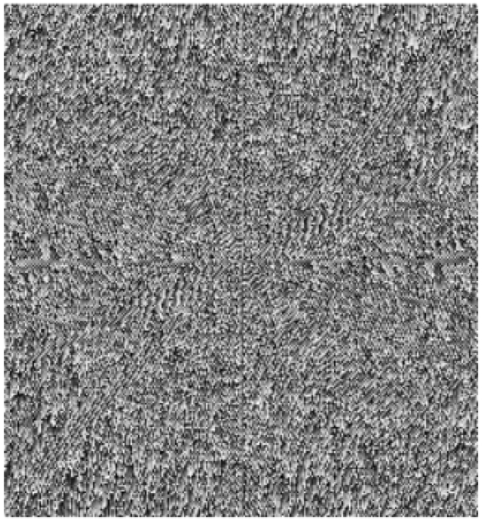
Centering: Looking at DFTs

Input Image

Log Amp of Centred DFT

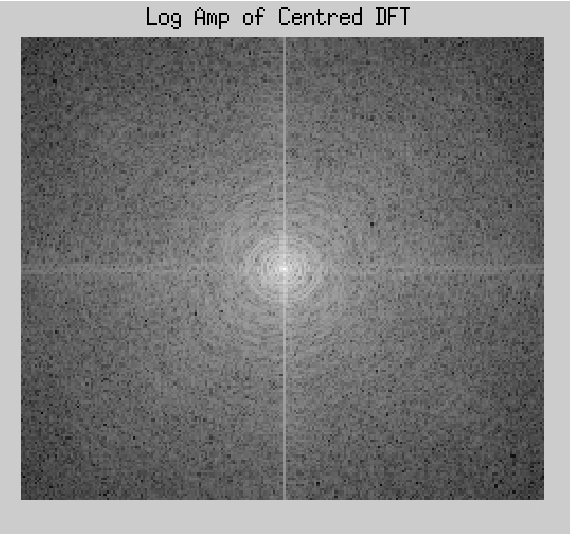


Phase of Centred DFT

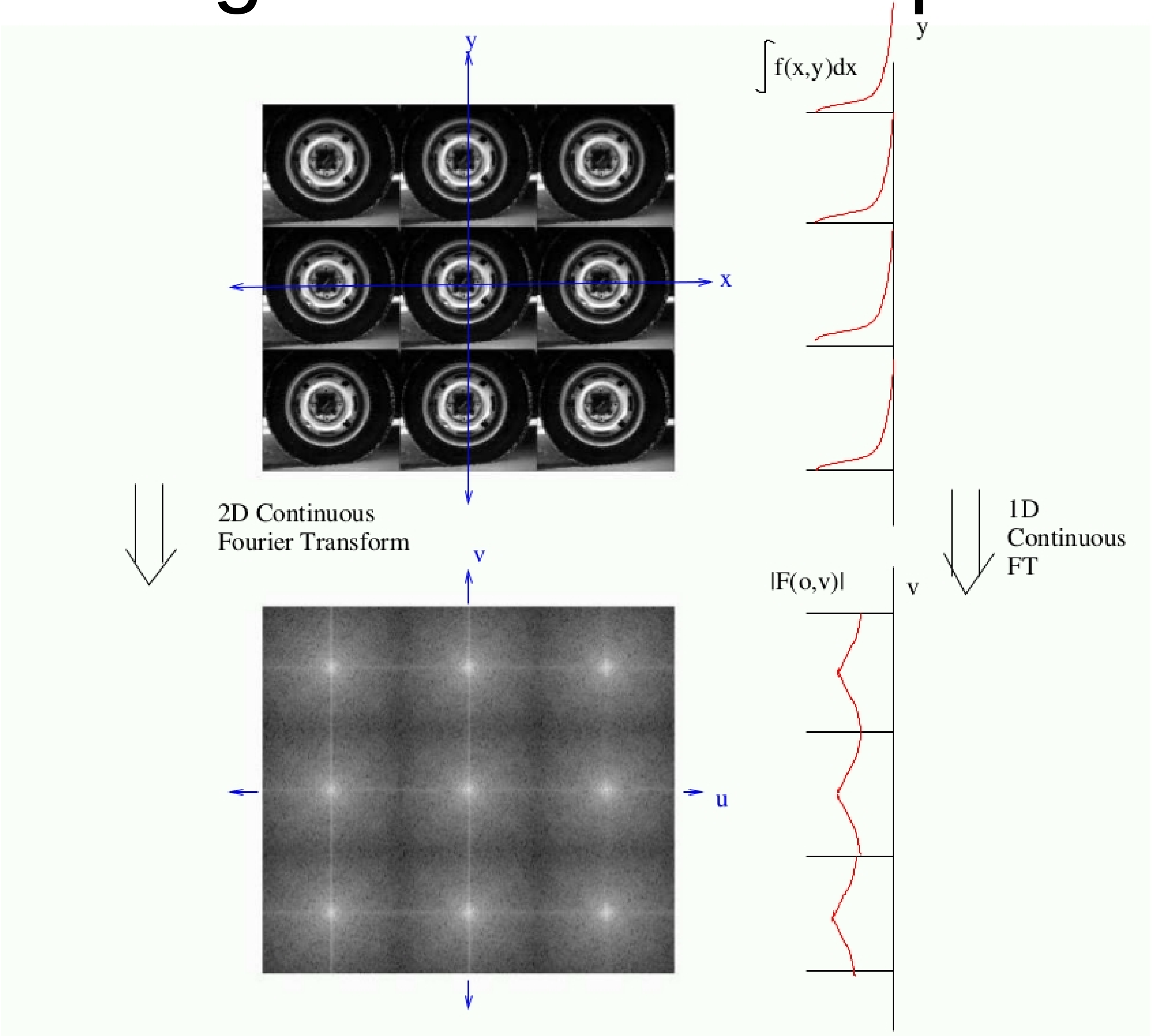


Edge effects: Example

DFT



Edge effects: Example



Edge effects

- Can get spikes or lines in FT because of sharp-edged objects in image, spike is perpendicular to direction of the edge.
- Can get large vertical spikes when there is a large difference in brightness between top and bottom of picture.
- Can get large horizontal spikes when there is a large difference in brightness between left and right.

2 min pause to discuss

- What does centering mean?
  Why is it useful?

- Give an example of edge effects.
  Explain why this happens.

Convolution

The convolution theorem is your friend!

$$\mathrm{DFT}(f * g) = \mathrm{DFT}(f) \cdot \ \mathrm{DFT}(\mathrm{g})$$

Convolution in spatial domain is equivalent to **multiplication** in frequency domain!

Filtering with DFT can be **much faster** than image filtering.

Convolution: Image vs DFT

A general linear convolution of $N_1 x N_1$ image with $N_2 x N_2$ convolving function (e.g. smoothing filter) requires in the **image domain** of order

$$N_1^2 N_2^2 \text{ operations.}$$

Instead using **DFT, multiplication, inverse DFT** one needs of order

$$4N^2 \mathrm{Log}_2 N \text{ operations.}$$

Here N is the smallest 2^n number greater or equal to N_1+N_2-1.

Conclusion: Use Image convolution for **small** convolving functions, and DFT for **large** convolving functions.

Convolution: Image vs DFT

Example 1: 10x10 pixel image, 5x5 averaging filter

Image domain: Num. of operations = $10^2 \times 5^2 = 2500$
Using DFT: $N_1+N_2-1=14$. Smallest 2^n is $2^4=16$.
Num. of operations = $4 \times 16^2 \times \log_2 16 = 4096$.
→ Use image convolution!

Example 2: 100x100 pixel image, 10x10 averaging filter

Image domain: Num. of operations = $100^2 \times 10^2 = 10^6$
Using DFT: $N_1+N_2-1=109$. Smallest 2^n is $2^7=128$.
Num of operations = $4 \times 128^2 \times \log_2 16 = 458752 \approx 5\times10^5$.
→ Use DFT convolution!

Convolution: Wrap-around errors

256x256 Image

Convolving Function

256x256 DFT Convolution

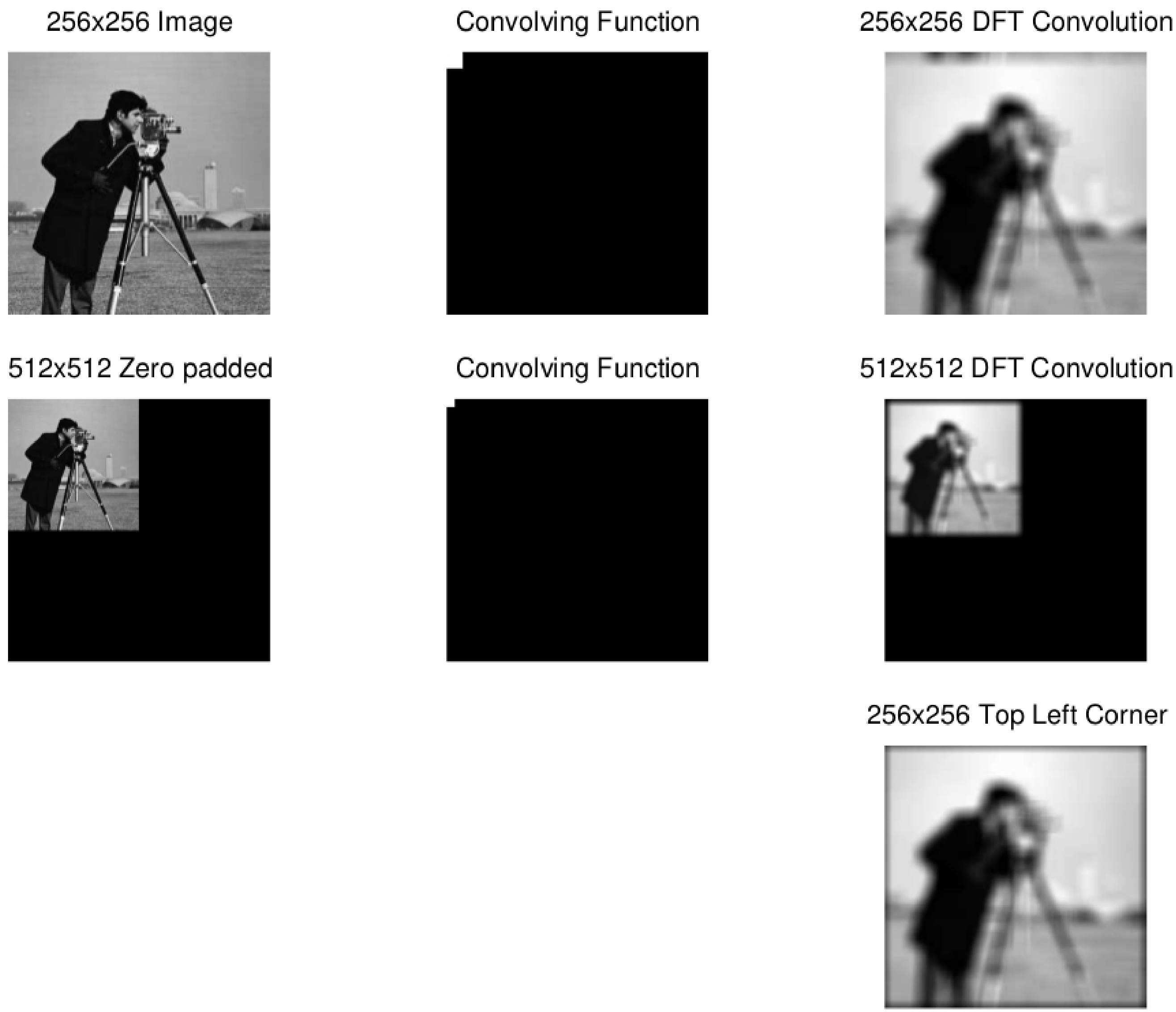
Convolution: Wrap-around errors

- Why? DFT assumes periodic images.
- Avoid by using *zero padding*! How much needed?
- Consider two NxM images. If image 1 is nonzero over region $N_1 x M_1$ and image 2 is nonzero over region $N_2 x M_2$ then we will not get any wrap-around errors if

$$N_1 + N_2 - 1 \leq N \qquad M_1 + M_2 - 1 \leq M.$$

If the above is not true we need to *zero-pad* the images to make the condition true!

Convolution: Wrap-around errors



Monday: Summary

- Brief repetition: What is 1D continuous FT
- The 2D Discrete Fourier Transform
- Important things in a discreet world:
  - Freq. Smoothing and leakage
  - Aliasing
  - Centering
  - Edge effects
  - Convolution
- **Two hours of Matlab exercises**