Lecture 4:
Image Transformations II:
Discrete Fourier Transform

Contents

1. Discrete Fourier Transform in 1-D
2. Discrete Fourier Transform in 2-D
3. Properties
4. Fast Fourier Transform

Announcement

- The next lecture (Thursday, Nov. 8) takes place in Lecture Hall 3 in the Mathematics Building E2.5, in the noon slot (12 am - 2 pm).
- It will be given by Dr. Andrés Bruhn.

Discrete Fourier Transform in 1-D

Goals:

- discrete analogue to the continuous Fourier transform
- should deal with sampled signals of *finite* extension
- signal with M values is decomposed into M frequency components

M I A

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Reminder from Lecture 3

- The continuous Fourier transform of a signal $f : \mathbb{R} \to \mathbb{R}$ with infinite extend is given by

$$\hat{f}(u) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi ux}\, dx \qquad (u \in \mathbb{R})$$

  with $i^2 = -1$.

- The corresponding inverse continuous Fourier transform is defined as

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(u)\, e^{i2\pi ux}\, du \qquad (x \in \mathbb{R})$$

Discrete Fourier Transform in 1-D (3)

Definition:

- The *discrete Fourier transform (DFT)* of a signal $f = (f_0, ..., f_{M-1})^\top$ with finite extend is given by

$$\hat{f}_p := \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} f_m \exp\left(-\frac{i2\pi pm}{M}\right) \qquad (p = 0, ..., M-1)$$

with $i^2 = -1$.

- The corresponding *inverse discrete Fourier transform* is defined as

$$f_m = \frac{1}{\sqrt{M}} \sum_{p=0}^{M-1} \hat{f}_p \exp\left(\frac{i2\pi pm}{M}\right) \qquad (m = 0, ..., M-1)$$

Discrete Fourier Transform in 1-D (4)

Interpretation as Change of Basis

- Let $f = (f_i)_{i=0}^{M-1}$ and $g = (g_i)_{i=0}^{M-1}$ be complex-valued vectors. We define the *Hermitian inner product* of f and g via

$$\langle f, g \rangle := \sum_{m=0}^{M-1} f_m \, \bar{g}_m$$

- One orthonormal basis of $(\mathbb{C}^M, \langle \cdot, \cdot \rangle)$ is given by the M vectors

$$v_p := \frac{1}{\sqrt{M}} \left(\exp\left(\frac{i2\pi p0}{M}\right), \exp\left(\frac{i2\pi p1}{M}\right), ..., \exp\left(\frac{i2\pi p(M-1)}{M}\right) \right)^\top$$

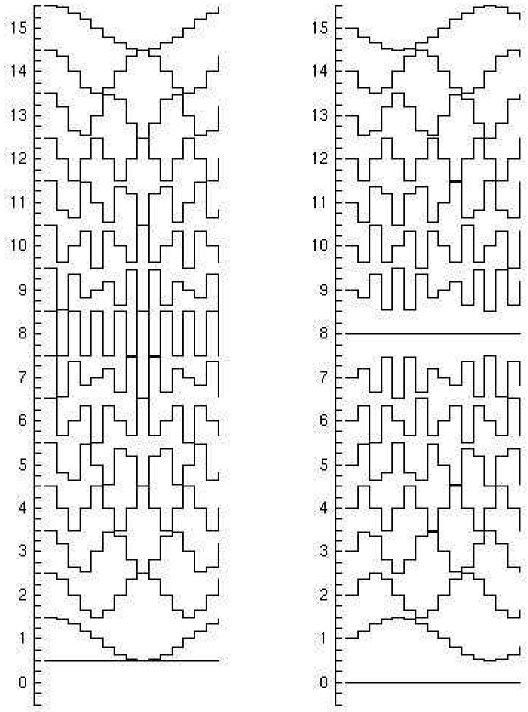
$$(p = 0, ..., M-1)$$

- Representing a vector f in this *(discrete) Fourier basis* yields

$$f = \sum_{p=0}^{M-1} \langle f, v_p \rangle \, v_p.$$

This is just the discrete Fourier transform with coefficients $\hat{f}_p := \langle f, v_p \rangle$.

Discrete Fourier Transform in 1-D (5)



Basis functions of the DFT for $M = 16$. **(a) Left:** Real part (cosine). **(b) Right:** Imaginary part (sine). Author: N. Khan (2005).

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Discrete Fourier Transform in 1-D (6)

Example: DFT of $f = (2, 3, 4, 4)^\top$

$$\hat{f}_0 = \frac{1}{2} \sum_{m=0}^{3} f_m \exp\left(-\frac{i 2\pi 0 m}{4}\right)$$

$$= \frac{1}{2}(f_0 + f_1 + f_2 + f_3)$$

$$= \frac{1}{2}\Big(2 + 3 + 4 + 4\Big) = \frac{13}{2}$$

$$\hat{f}_1 = \frac{1}{2} \sum_{m=0}^{3} f_m \exp\left(-\frac{i 2\pi 1 m}{4}\right)$$

$$= \frac{1}{2}\Big(f_0\, e^0 + f_1\, e^{-i\pi/2} + f_2\, e^{-i\pi} + f_3\, e^{-i3\pi/2}\Big)$$

$$= \frac{1}{2}\Big(2 \cdot 1 + 3 \cdot (-i) + 4 \cdot (-1) + 4i\Big) = \frac{-2 + i}{2}$$

$$
\begin{aligned}
\hat{f}_2 &= \frac{1}{2}\sum_{m=0}^{3} f_m \exp\left(-\frac{i2\pi 2m}{4}\right) \\
&= \frac{1}{2}\left(2\,e^0 + 3\,e^{-i\pi} + 4\,e^{-i2\pi} + f_3\,e^{-i3\pi}\right) \\
&= \frac{1}{2}\Big(2\cdot 1 + 3\cdot(-1) + 4\cdot 1 + 4\cdot(-1)\Big) \;=\; -\frac{1}{2}
\end{aligned}
$$

$$
\begin{aligned}
\hat{f}_3 &= \frac{1}{2}\sum_{m=0}^{3} f_m \exp\left(-\frac{i2\pi 3m}{4}\right) \\
&= \frac{1}{2}\Big(2\,e^0 + 3\,e^{-i3\pi/2} + 4\,e^{-i3\pi} + 4\,e^{-i9\pi/2}\Big) \\
&= \frac{1}{2}\Big(2\cdot 1 + 3\cdot(i) + 4\cdot(-1) + 4(-i)\Big) \;=\; \frac{-2-i}{2}
\end{aligned}
$$

If you would like to acquire more experience with the DFT, compute the inverse transformation.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Remarks

The previous example illustrates some general properties of the DFT:

- $\hat{f}_0 = \frac{1}{\sqrt{M}}\sum_{m=0}^{M-1} f_m$ is $\sqrt{M}$ times the average grey value $\frac{1}{M}\sum_{m=0}^{M-1} f_m$.
- $\mathrm{Re}(\hat{f}_p)$ is an even function in p with respect to $M/2$:

$$\mathrm{Re}(\hat{f}_1) \;=\; \mathrm{Re}(\hat{f}_3)$$

- $\mathrm{Im}(\hat{f}_p)$ is an odd function in p with respect to $M/2$:

$$\mathrm{Im}(\hat{f}_1) \;=\; -\mathrm{Im}(\hat{f}_3).$$

  It vanishes for $p = M/2$:

$$\mathrm{Im}(\hat{f}_2) \;=\; 0.$$

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Discrete Fourier Transform in 2-D

Definition:

- The *discrete Fourier transform* of an image $f = (f_{m,n})$ with $m = 0, ..., M-1$ and $n = 0, ..., N-1$ is given by

$$\hat{f}_{p,q} \;:=\; \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_{m,n} \exp\left(-\frac{i2\pi pm}{M}\right) \exp\left(-\frac{i2\pi qn}{N}\right)$$
$$(p = 0, ..., M-1; \;\; q = 0, ..., N-1).$$

- The corresponding *discrete inverse Fourier transform* in 2-D is defined as

$$f_{n,m} \;=\; \frac{1}{\sqrt{MN}} \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \hat{f}_{p,q} \exp\left(\frac{i2\pi pm}{M}\right) \exp\left(\frac{i2\pi qn}{N}\right)$$
$$(n = 0, ..., M-1; \;\; m = 0, ..., N-1).$$

The DFT in higher dimensions is defined in an analogue way.
Just like the continuous FT, the DFT is separable.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Properties of the Discrete Fourier Transform

Many important properties of the continuous FT carry over to the discrete FT:

- linearity
- shift theorem
- convolution theorem

Some properties, however, can only be approximated on a discrete grid:

- scaling theorem
- rotation invariance

Often one uses the continuous FT for designing filters,
and the discrete FT for implementing them.

Properties of the Discrete Fourier Transform (2)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Important Difference to the Continuous FT

- The signal f has finite extension: $f_0, \ldots, f_{M-1}$.
- The periodicity of the complex exponential function creates a periodic continuation of the signal in its Fourier representation:

$$\hat{f}_{p,q} = \hat{f}_{p+kM,\, q+lN}$$

  This creates undesired boundary artifacts.
- Example 1: Discontinuities at periodically extended boundaries create high-frequent Fourier components in x- and y-direction.
- Example 2: *wraparound errors* in connection with convolutions:
  Grey values near the right boundary perturb grey values at the left boundary.

Properties of the Discrete Fourier Transform (3)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

How can Wraparound Errors be Handled ?

- **Fatalism**
  Do nothing and trust only your results far away from the boundaries.
  Disadvantage: Not many results are reliable under large convolution kernels.
- **Zero Padding**
  Supplement a layer of zeroes at the boundaries whose thickness respects the size of the convolution kernel.
  Disadvantage: Also zeroes can spoil your signal !
- **Mirror Image at Boundaries**
  cleanest solution
  Disadvantage: The computational load in each dimension is doubled.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Translation of the Fourier Spectrum

- Problem: DFT yields segment with frequencies in $[0, M-1] \times [0, N-1]$.
  It would be nice to shift the origin of the spectrum to the centre $(M/2, N/2)$.

- discrete shift theorem gives transformation pairs

$$f_{m-m_0,\, n-n_0} \iff \hat{f}_{p,q} \exp\left(-\tfrac{i2\pi p m_0}{M}\right) \exp\left(-\tfrac{i2\pi q n_0}{N}\right)$$

$$f_{m,n} \exp\left(\tfrac{i2\pi p_0 m}{M}\right) \exp\left(\tfrac{i2\pi q_0 n}{N}\right) \iff \hat{f}_{p-p_0,\, q-q_0}$$

- With $p_0 = M/2$ and $q_0 = N/2$, one replaces the image $f_{m,n}$ by

$$f_{m,n} \exp\left(\frac{i\pi M m}{M}\right) \exp\left(\frac{i\pi N n}{N}\right) = f_{m,n}(-1)^{m+n}.$$

  All one has to do is to superpose a checkerboard-like sign pattern.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Logarithmic Scaling of the Fourier Spectrum

- Problem: dynamic range of the Fourier spectrum covers many orders of magnitude

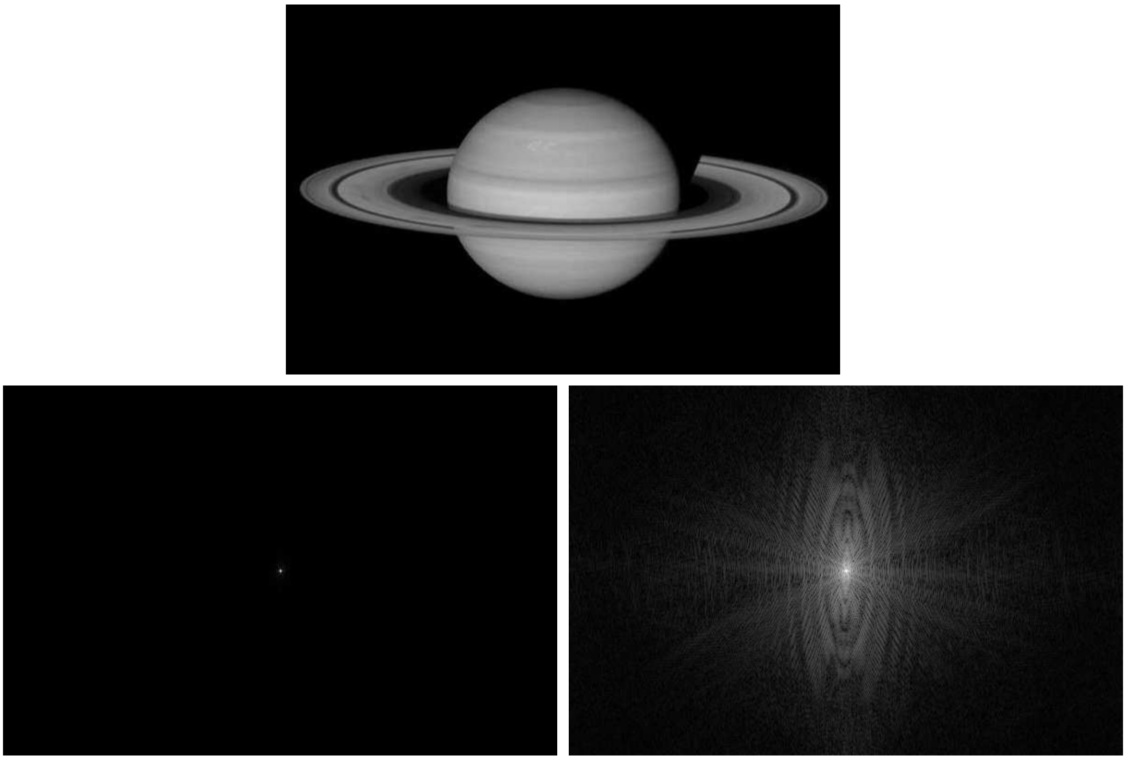
- For visualisation purposes it is therefore common to use a logarithmic transformation:

$$D_{p,q} = c \log\left(1 + |\hat{f}_{p,q}|\right).$$

- Often c is chosen such that

$$\max_{p,q} D_{p,q} = 255.$$

  This allows convenient bytewise coding of the result, since its range is in $[0, 255]$.



(a) Top: Image of the planet Saturn, 600×400 pixels (`www.androidworld.com/Saturn.jpeg`).
(b) Bottom left: Fourier spectrum. **(c) Bottom right:** Fourier spectrum after logarithmic scaling. with $\max D_{p,q} = 255$. Author: J. Weickert (2005).

MIA 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

The Fast Fourier Transform (FFT)

(Gauß 1805; Cooley/Tukey 1965)

- Litteral implementation of 1-D DFT of a signal of length M is quite expensive: M^2 (complex) multiplications and $M^2 - M$ (complex) additions
- Basic idea behind the *Fast Fourier Transform (FFT)*: divide-and-conquer
  - split problem of size M into two subproblems of size $M/2$
  - continue until size 1 is reached
- Advantages:
  - very efficient: $\mathcal{O}(M \log_2 M)$ operations
  - available in many numerical packages (see e.g. `www.fftw.org`)
- Disadvantage:
  - standard FFT requires signals of size $M = 2^k$
- For images one exploits the separability of the DFT:
  - hardly additional memory requirements
  - well-suited for parallel computing

Fast Fourier Transform (2)


(a) Left: John Wilder Tukey (1915–2000) did not only popularise the FFT in 1965 (with James Cooley), he also poineered robust statistics (including e.g. median filtering) and coined the words *bit* and *software*. Source: `http://www-history.mcs.st-and.ac.uk/PictDisplay/Tukey.html`. **(b) Right:** Carl Friedrich Gauß (1777–1855) is often considered as one of the greatest mathematicians of all times. He already used the FFT for astronomical computations in 1805. Source: `http://de.wikipedia.org/wiki/Bild:Carl_Friedrich_Gauss.jpg`.

Fast Fourier Transform (3)

The Inverse FFT

- For computing the inverse FFT, no second algorithm is needed:
  - Just replace the Fourier coefficients $\hat{f}_p$ by their complex conjugates $\bar{\hat{f}}_p$.
  - Apply the FFT to these numbers.
- Explanation: Computing the inverse FT

$$f_m = \frac{1}{\sqrt{M}} \sum_{p=0}^{M-1} \hat{f}_p \exp\left(\frac{i2\pi pm}{M}\right)$$

and taking its complex conjugate gives

$$\bar{f}_m = \frac{1}{\sqrt{M}} \sum_{p=0}^{M-1} \bar{\hat{f}}_p \exp\left(-\frac{i2\pi pm}{M}\right).$$

Moreover, $\bar{f}_m = f_m$ for real-valued images.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Summary

- The discrete Fourier transform (DFT) decomposes a discrete signal of size M in M frequency components.
- similar properties as continuous FT:
complex-valued, linear, separable, shift theorem, convolution theorem
- main difference: finite signal size introduces periodicity
- creates problems such as wraparound errors
- very efficient numerical algorithm: Fast Fourier Transform (FFT)
- complexity of $\mathcal{O}(M \log_2 M)$ in 1-D

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Literature

- R. C. Gonzalez, R. E. Woods: *Digital Image Processing*. Prentice Hall, Upper Saddle River, Second Edition, 2002.
(Chapter 4 gives an extended description of many technicalities.)
- K. R. Castleman: *Digital Image Processing*. Prentice Hall, Upper Saddle River, 1996.
(One of the best textbooks on linear concepts such as the Fourier transform, convolutions and linear systems.)
- R. Bracewell: *The Fourier Transform and its Applications*. McGraw-Hill, New York, 2002.
(A classical reference on the Fourier transform.)
- D. N. Rockmore: The FFT: An algorithm the whole family can use. *Computing in Science and Engineering*, Vol. 2, No. 1, pp. 60–64, 2000.
(An entertaining article on the past, the presence and the future of the FFT.)

Assignment P1 – Programming

Please download the required files from the webpage

`http://www.mia.uni-saaland.de/Teaching/ipcv07.shtml`

into your own directory. You can unpack them with the command `tar xvzf Ex01.tgz`.

Problem 1 (Noise Generation and Convolution) (6 points)

The file `gauss_noise.c` implements a programme that adds Gaussian noise of mean 0 and standard deviation σ to an input image and writes out the resulting output image to disc. Supplement this file with the missing code for the Box-Muller method. The programme can be compiled using the command

```
gcc -O2 -o gauss_noise gauss_noise.c -lm
```

(a) Use the test image `couple.pgm` and add Gaussian noise of standard deviation $\sigma = 10$, 20 and 40. Compare the results visually by displaying them via the command `xv image_name.pgm &`. How does the standard deviation of the noisy images change if you increase the noise level?

(b) Use the precompiled programme `gauss_conv` to convolve the noisy images with a Gaussian kernel of standard deviation σ. This can be done by typing `./gauss_conv`. Which are suitable values for σ so that the noise is significantly reduced but the images still look reasonable? How does the standard deviation change after convolution ?

Problem 2 (Interpretation of the Fourier Spectrum) (6 points)

The programme `fourierspectrum` computes the logarithmically transformed Fourier spectrum $c\ln(1 + \hat{f}(u, v))$ of an image $f(x, y)$ by means of the FFT. The lowest frequencies have been shifted towards the centre of the image.
Apply this programme to the images `pattern.pgm`, `gauss1.pgm`, `gauss2.pgm`, `gauss3.pgm` and `tile.pgm` by typing `./fourierspectrum`, and visualise them by using `xv image_name.pgm &`.

(a) Why do you observe a three-point spectrum for `pattern.pgm` and why it is located this way ?

(b) Why is the DFT of `gauss.pgm` not rotationally symmetric ?

(c) Can you find aliasing artifacts in the spectrum of `tile.pgm` ?
(This image has been downsampled with `xv` to half its size.)

Assignment P1 (3)

Problem 3 (Filtering in the Fourier Domain) (8 points)

With the subroutine filter.c you can now influence the image in the Fourier domain. By compiling the result with

```
gcc -O2 -o FFT filter.c FFT.o -lm
```

you create an executable programme FFT, which performs a Fast Fourier Transform, executes your ideas coded in filter.c, and applies an inverse FFT afterwards.

(a) The image dancing.pgm has been scanned from a newspaper. It reveals a periodic grid raster which looks unpleasant. Can you remove this by a suitable filter in the Fourier domain ? (Pressing the middle mouse button under xv gives the pixel coordinates.)

(b) Try to devise another suitable filter that removes the dominating lines in the image tile.pgm.

Assignment P1 (4)

Submission

Please remember that up to three people from the same tutorial group can work and submit their results together. For submitting the files rename the main directory Ex01 to Ex01_<your_name> and use the command

```
tar czvf Ex01_<your_name>.tgz Ex01_<your_name>
```

to pack the data. The directory that you pack and submit should at least contain the following files:

- the source code for gauss_noise.c with the implemented Box-Muller algorithm
- the three noisy and denoised (convolved) variants of the image couple.pgm
- the source code for both versions of filter.c (for the images dancing.pgm and tile.pgm)
- the filtered versions of dancing.pgm and tile.pgm
- a text file README that contains answers to all questions in the problems 1 and 2 as well as information on all people working together for this assignment.

Please make sure that only your final version of the programmes and images are included. Submit the file via e-mail to your tutor via the address:

```
ipcv-xx@mia.uni-saarland.de
```

where xx is either t1, t2, t3, t4, w1 or w2 depending on your tutorial group.

Deadline for submission: Tuesday, November 13, 10 am (before the lecture)