Lecture 6:
Image Transformations III:
Discrete Cosine Transform and Image Pyramids

Contents

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 27 28 29 30 31 32 33 34 35 36

Motivation

- The discrete Fourier transform makes the frequency content of images explicit. However, it suffers from some drawbacks:
  - (a) It involves complex-valued computations.
    Is there a similar, real-valued transformation?
  - (b) Boundary artifacts may arise due to the automatic periodic extension.
    Mirroring the image as a remedy leads to a higher computational burden.
    Is there a transformation that performs mirroring automatically?
  - (c) Spatially localised image structures affect all Fourier coefficients globally.
    Are there frequency-like decompositions with better spatial localisation?

- Let us study two transformations that address these drawbacks:
  - The discrete cosine transform addresses (a) and (b).
  - Image pyramids avoid also drawback (c).

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 27 28 29 30 31 32 33 34 35 36

Discrete Cosine Transform in 1-D

Choosing the Inner Product and the Basis

- We want a real-valued transform that makes the frequency content explicit.
- Thus, we can consider the Euclidean inner product.
  For two vectors $\boldsymbol{f} = (f_i)_{i=0}^{M-1}$ and $\boldsymbol{g} = (g_i)_{i=0}^{M-1}$ in $\mathbb{R}^M$ it is given by

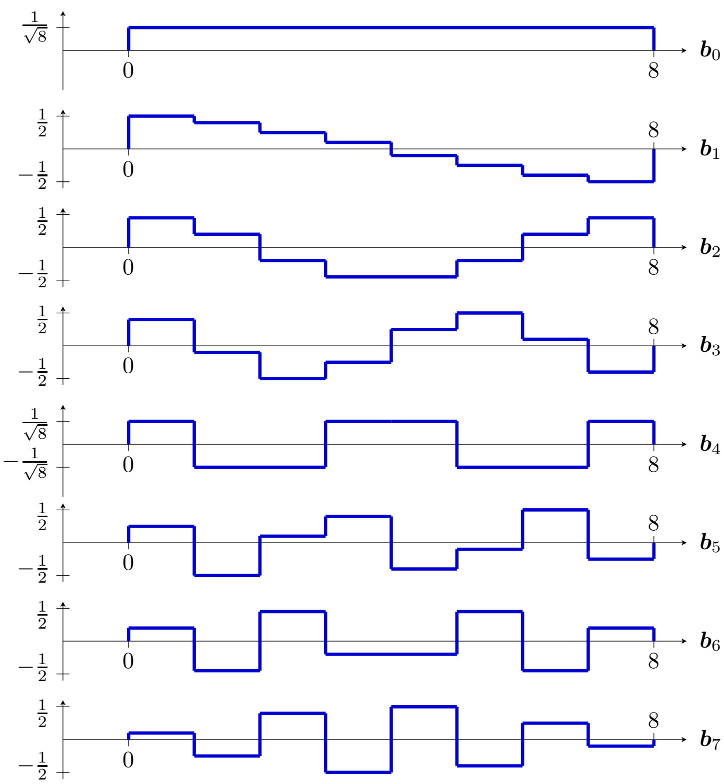
$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle := \sum_{m=0}^{M-1} f_m \, g_m \, .$$

- As orthonormal basis of $(\mathbb{R}^M, \langle \cdot, \cdot \rangle)$ we choose the M vectors

$$\boldsymbol{b}_p := c_{p,M} \left(\cos\left(\tfrac{\pi \, 1 \, p}{2M}\right), \cos\left(\tfrac{\pi \, 3 \, p}{2M}\right), \ldots, \cos\left(\tfrac{\pi \, (2M-1) \, p}{2M}\right) \right)^\top \quad (p = 0, \ldots, M-1).$$

with $c_{0,M} := \sqrt{\frac{1}{M}}$, and with $c_{p,M} := \sqrt{\frac{2}{M}}$ for $p > 0$.

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 27 28 29 30 31 32 33 34 35 36



Visualisation of the basis vectors $\boldsymbol{b}_0,\ldots,\boldsymbol{b}_7$ for $M = 8$. Author: T. Schneevoigt.

Discrete Cosine Transform in 1-D (3)

Discrete Cosine Transform (DCT)

- The DCT computes the coefficients $\tilde{f}_p = \langle \boldsymbol{f}, \boldsymbol{b}_p \rangle$ for representing $\boldsymbol{f}$ in our basis.
- For a 1-D signal $\boldsymbol{f} = (f_0, ..., f_{M-1})^\top$ it is given by

$$\tilde{f}_p \; := \; \sum_{m=0}^{M-1} f_m \, c_{p,M} \, \cos\left(\frac{\pi\,(2m+1)\,p}{2M}\right) \qquad (p = 0, ..., M-1)$$

with $c_{0,M} := \sqrt{\frac{1}{M}}$, and with $c_{p,M} := \sqrt{\frac{2}{M}}$ for $p > 0$.

Inverse Discrete Cosine Transform

- The inverse DCT assembles $\boldsymbol{f}$ from its DCT coefficients $\tilde{f}_p$ via $\boldsymbol{f} = \sum_{p=0}^{M-1} \tilde{f}_p \, \boldsymbol{b}_p$.
- For a 1-D signal $\tilde{\boldsymbol{f}} = (\tilde{f}_0, ..., \tilde{f}_{M-1})^\top$ it is given by

$$f_m \; := \; \sum_{p=0}^{M-1} \tilde{f}_p \, c_{p,M} \, \cos\left(\frac{\pi\,(2m+1)\,p}{2M}\right) \qquad (m = 0, ..., M-1)\,.$$

Discrete Cosine Transform in 1-D (4)

Example: DCT of $\boldsymbol{f} = (6, 4, 5, 1)^\top$

For $M = 4$ we have the following DCT basis vectors:

$$\begin{aligned}
\boldsymbol{b}_0 \;&=\; \tfrac{1}{2}\Big(\cos\left(\tfrac{\pi}{8}\,0\right),\ \cos\left(\tfrac{3\pi}{8}\,0\right),\ \cos\left(\tfrac{5\pi}{8}\,0\right),\ \cos\left(\tfrac{7\pi}{8}\,0\right)\Big)^\top \\
&=\; \Big(0.5,\ 0.5,\ 0.5,\ 0.5\Big)^\top, \\
\boldsymbol{b}_1 \;&=\; \tfrac{1}{\sqrt{2}}\Big(\cos\left(\tfrac{\pi}{8}\,1\right),\ \cos\left(\tfrac{3\pi}{8}\,1\right),\ \cos\left(\tfrac{5\pi}{8}\,1\right),\ \cos\left(\tfrac{7\pi}{8}\,1\right)\Big)^\top \\
&\approx\; \Big(0.653,\ 0.271,\ -0.271,\ -0.653\Big)^\top, \\
\boldsymbol{b}_2 \;&=\; \tfrac{1}{\sqrt{2}}\Big(\cos\left(\tfrac{\pi}{8}\,2\right),\ \cos\left(\tfrac{3\pi}{8}\,2\right),\ \cos\left(\tfrac{5\pi}{8}\,2\right),\ \cos\left(\tfrac{7\pi}{8}\,2\right)\Big)^\top \\
&=\; \Big(0.5,\ -0.5,\ -0.5,\ 0.5\Big)^\top, \\
\boldsymbol{b}_3 \;&=\; \tfrac{1}{\sqrt{2}}\Big(\cos\left(\tfrac{\pi}{8}\,3\right),\ \cos\left(\tfrac{3\pi}{8}\,3\right),\ \cos\left(\tfrac{5\pi}{8}\,3\right),\ \cos\left(\tfrac{7\pi}{8}\,3\right)\Big)^\top \\
&\approx\; \Big(0.271,\ -0.653,\ 0.653,\ -0.271\Big)^\top.
\end{aligned}$$

Discrete Cosine Transform in 1-D (5)

With the Euclidean inner product, the DCT coefficients of the signal $\boldsymbol{f} = (6,4,5,1)^\top$ are given by

$$\begin{aligned}
\tilde{f}_0 &= \langle \boldsymbol{f}, \boldsymbol{b}_0 \rangle = \left(6 \cdot 0.5 + 4 \cdot 0.5 + 5 \cdot 0.5 + 1 \cdot 0.5\right) = 8,\\
\tilde{f}_1 &= \langle \boldsymbol{f}, \boldsymbol{b}_1 \rangle = \left(6 \cdot 0.653 + 4 \cdot 0.271 - 5 \cdot 0.271 - 1 \cdot 0.653\right) = 2.994,\\
\tilde{f}_2 &= \langle \boldsymbol{f}, \boldsymbol{b}_2 \rangle = \left(6 \cdot 0.5 - 4 \cdot 0.5 - 5 \cdot 0.5 + 1 \cdot 0.5\right) = -1,\\
\tilde{f}_3 &= \langle \boldsymbol{f}, \boldsymbol{b}_3 \rangle = \left(6 \cdot 0.271 - 4 \cdot 0.653 + 5 \cdot 0.653 - 1 \cdot 0.271\right) = 2.008.
\end{aligned}$$

Note that $\tilde{f}_0$ contains the rescaled average grey value of $\boldsymbol{f}$.

By plugging in, it is not hard to check that the inverse DCT recovers $\boldsymbol{f}$:

$$\boldsymbol{f} = \tilde{f}_0 \boldsymbol{b}_0 + \tilde{f}_1 \boldsymbol{b}_1 + \tilde{f}_2 \boldsymbol{b}_2 + \tilde{f}_3 \boldsymbol{b}_3.$$

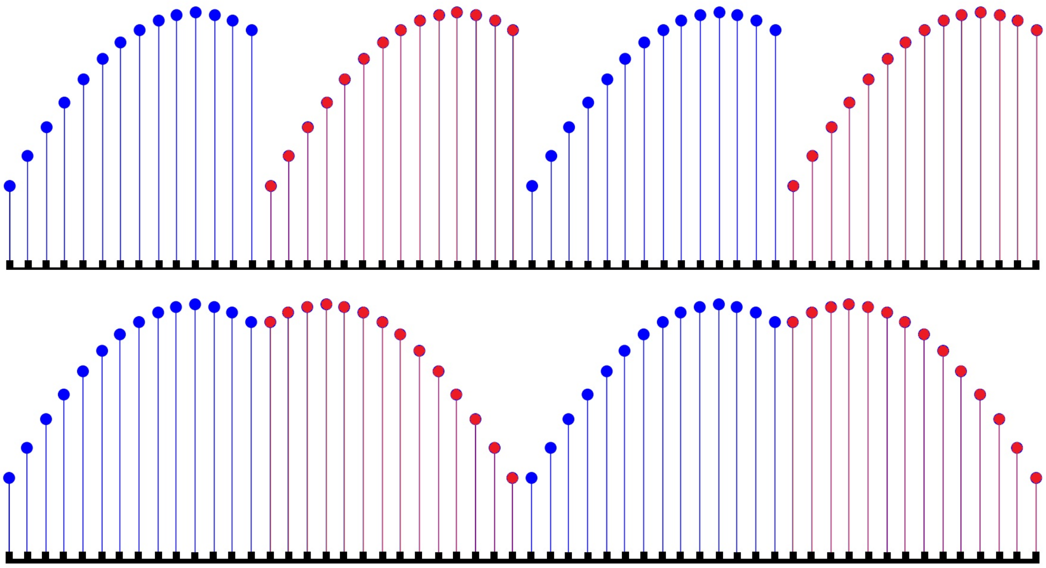
Discrete Cosine Transform in 1-D (6)

Properties

- In contrast to the DFT, the DCT allows computations with real-valued instead of complex-valued coefficients.
- Homework Assignment H3, Problem 2 helps you to understand the DCT:
  - The DCT is essentially a DFT of the signal, extended by mirroring.
  - Periodic extensions of the mirrored signal are even.
  - Thus, its DFT does not require (odd) sine components and is real-valued.
- The mirrored signal extension has one important benefit:
  - The DCT suffers less from boundary artifacts than the DFT (which has discontinuities in the periodic extension of uneven signals).
  - Therefore, its coefficients decrease more rapidly towards high frequencies.
- Efficient implementations in software and hardware are possible, similar to the FFT.
- Unfortunately, the DCT does not enjoy some key benefits of the FT and the DFT:
  - Convolutions and derivatives cannot be computed in a similar nice way.
  - Thus, it is less popular for filter design and signal analysis.

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 27 28 29 30 31 32 33 34 35 36

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 27 28 29 30 31 32 33 34 35 36



Top: The DFT extends the signal periodically. This can lead to discontinuities that result in high frequency artifacts in the Fourier domain. **Bottom:** The DCT extends the signal by mirroring. This creates a continuous extension and a more rapid decay towards high frequencies. Author: A. Goswami.

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 27 28 29 30 31 32 33 34 35 36

Discrete Cosine Transform in 2-D

- Consider a discrete image $\boldsymbol{f} = (f_{m,n})$ with $m = 0, ..., M-1$ and $n = 0, ..., N-1$.
  Its *discrete cosine transform* is given by

$$\tilde{f}_{p,q} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f_{m,n}\, c_{p,M}\, c_{q,N} \cos\left(\frac{\pi\,(2m+1)\,p}{2M}\right) \cos\left(\frac{\pi\,(2n+1)\,q}{2N}\right)$$

$$(p = 0, ..., M-1;\;\; q = 0, ..., N-1)\,.$$

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

$$f_{m,n} = \sum_{p=0}^{M-1}\sum_{q=0}^{N-1} \tilde{f}_{p,q}\, c_{p,M}\, c_{q,N} \cos\left(\frac{\pi\,(2m+1)\,p}{2M}\right) \cos\left(\frac{\pi\,(2n+1)\,q}{2N}\right)$$

$$(m = 0, ..., M-1;\;\; n = 0, ..., N-1)\,.$$

- Any multidimensional DCT is separable into 1-D DCTs.
  Thus, it is sufficient to have the 1-D definition and its implementation.

Discrete Cosine Transform in 2-D (2)

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 27 28 29 30 31 32 33 34 35 36

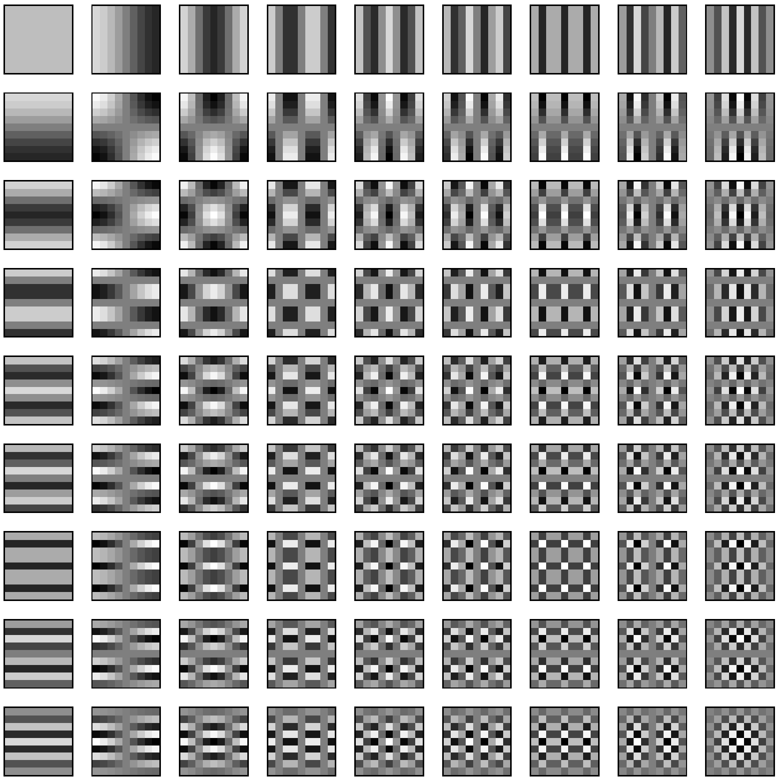


Illustration of the 81 basis vectors of the 2-D DCT for $M = N = 9$. Author: T. Schneevoigt.

Discrete Cosine Transform in 2-D (3)

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 27 28 29 30 31 32 33 34 35 36

Main Application of the DCT

- The rapid decay of the DCT coefficients towards high frequencies is particularly useful for lossy image and video compression (e.g. in JPEG):
  - Only the few coefficients with large magnitude determine the visual impression.
  - They can be quantised in a coarse manner, and the others are neglected.
  - More details will be given in Lecture 8.
- The DCT is superior to the DFT when it comes to image compression.
  The DFT is easier to handle for fast convolutions and filter analysis.

Left: Original image, 256×256 pixels. **Middle:** Lower right quadrant of the logarithmically rescaled spectrum of the DFT. The origin is in the upper left corner. Note the well visible coefficients from the periodic extension, in particular along the y-axis. **Right:** Spectrum of the logarithmically rescaled DCT coefficients. Due to the extension by mirroring, artifacts are greatly reduced. Author: J. Weickert.

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 27 28 29 30 31 32 33 34 35 36

Gaussian Pyramids

Goals

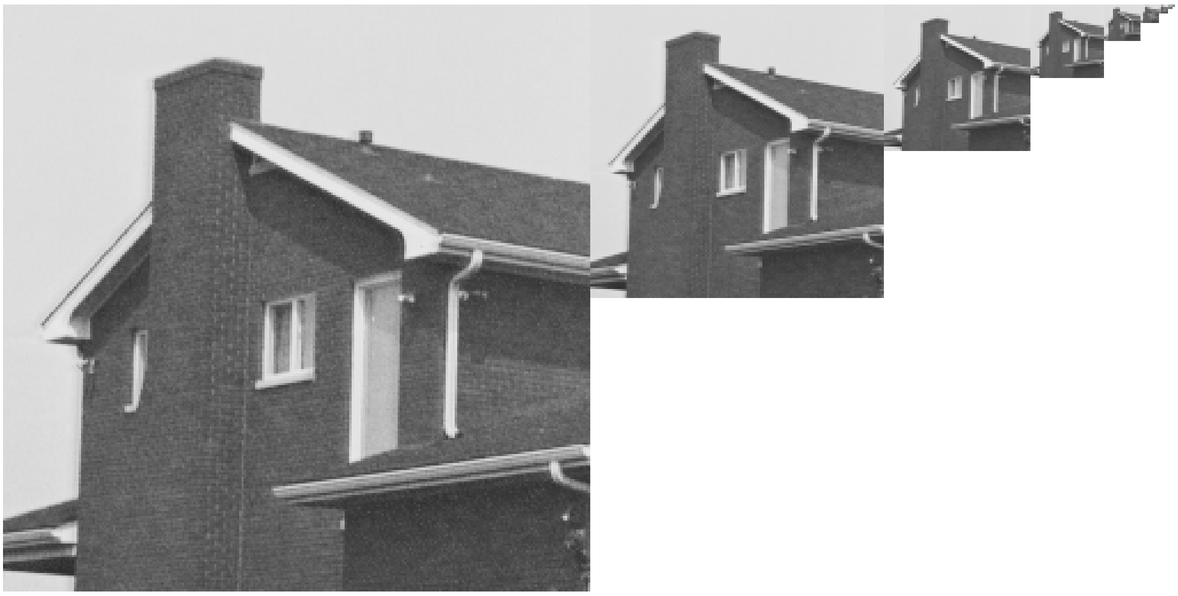
- image representation with multiple resolution levels
- going to the next coarser resolution should reduce the image size in each dimension by a factor 2.

How to Reduce a Signal to Half its Size

- Sampling theorem (Lecture 5):
  Attenuate high frequences in order to reduce aliasing effects.
- Thus, first smooth the signal by averaging, then sample with half the frequency.
- simplest smoothing mask in 1-D averages two neighbouring pixels:

$$\left(\tfrac{1}{2},\ \tfrac{1}{2}\right)$$

- in m dimensions: apply this mask subsequently along all m axes (separability)



Gaussian pyramid of an image of size 256×256. Author: J. Weickert.

Formal Description of the 1-D Gaussian Pyramid

- consider 1-D signal $\boldsymbol{u} = (u_0, ..., u_{2^N-1})^\top$ of length 2^N ("signal at level N")
- signal at level k should have 2^k values
- Define a *restriction operator* from level k to level $k-1$ as multiplication with the $2^{k-1} \times 2^k$ matrix

$$\boldsymbol{R}_k^{k-1} := \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ \multicolumn{13}{c}{\dotfill} \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

- The *Gaussian pyramid (Gauß–Pyramide)* $\{\boldsymbol{v}^N, ..., \boldsymbol{v}^0\}$ of $\boldsymbol{u}$ is defined as

$$\begin{aligned} \boldsymbol{v}^N &:= \boldsymbol{u}, \\ \boldsymbol{v}^{k-1} &:= \boldsymbol{R}_k^{k-1} \boldsymbol{v}^k \qquad (k = N, ..., 1). \end{aligned}$$

Gaussian Pyramids (4)

Complexity Aspects

- The Gaussian pyramid requires hardly more disk space than the original image: Since $\sum_{k=0}^{\infty} q^k = \frac{1}{1-q}$ for $|q| < 1$, the factors are bounded from above as follows:

$$\text{in 1-D:} \quad 1 + \frac{1}{2} + \frac{1}{4} + \ldots = \frac{1}{1-\frac{1}{2}} = 2$$

$$\text{in 2-D:} \quad 1 + \frac{1}{4} + \frac{1}{16} + \ldots = \frac{1}{1-\frac{1}{4}} = \frac{4}{3}$$

$$\text{in 3-D:} \quad 1 + \frac{1}{8} + \frac{1}{64} + \ldots = \frac{1}{1-\frac{1}{8}} = \frac{8}{7}$$

- The pyramid decomposition has optimal complexity: linear!
- There even exist hardware realisations of image pyramids.
- Unfortunately, pyramids are not invariant under translations.

Gaussian Pyramids (5)

Coarse-to-Fine Strategies

- Many image processing methods are iterative.
  A good initialisation leads to fast convergence.
- widely used strategy to obtain good initialisations:
  - apply iterative method at a coarse level of the Gaussian pyramid first
  - gives rapidly a coarse scale approximation to the fine scale solution
  - interpolate it and use it as initialisation at the next finer scale
- additional advantages:
  - coarser scales are less affected by noise
  - also useful in optimisation methods with many local optima:
    avoids getting trapped in irrelevant local optima
- related technique in numerical analysis: *multigrid methods*
  (allow e.g. to solve linear systems of equations in linear complexity)

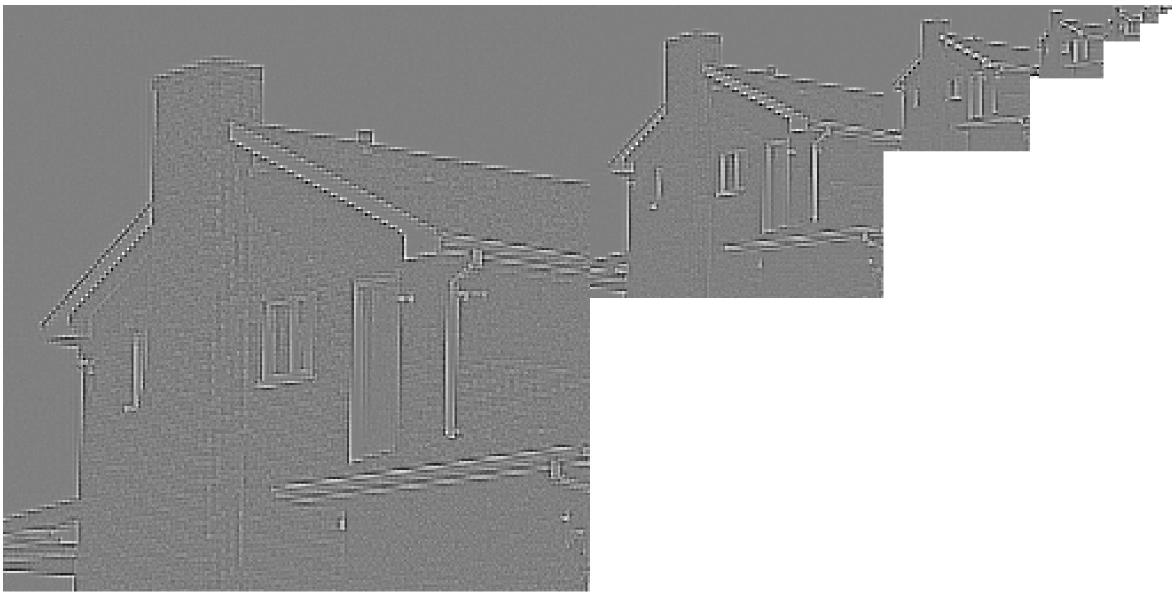
Laplacian Pyramids

Goals

- decompose an image in its *spatial domain (!)* in different frequency bands
- alternative to Fourier analysis that requires to work in the *frequency domain*

Basic Idea

- Downsampling in the Gaussian pyramid damps higher frequencies, while lower frequencies may pass *(lowpass filter).*
- Subtracting subsequent levels singles out certain frequency bands *(bandpass filter).*
- To subtract images of equal size, we interpolate the downsampled image. Simplest interpolation strategy: upscaling by pixel doubling (separable).
- This pyramid of difference images is called *Laplacian pyramid.*



Laplacian pyramid of an image of size 256×256. The grey values of the levels ≥ 1 have been rescaled by a factor 1.59 and shifted by 127.5 to improve visibility. One can still see aliasing artifacts due to the relatively poor smoothing effect of the kernel $(\frac{1}{2}, \frac{1}{2})$ in the Gaussian pyramid. Author: J. Weickert.

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 27 28 29 30 31 32 33 34 35 36

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 27 28 29 30 31 32 33 34 35 36

Formal Description of the 1-D Laplacian Pyramid

- Given: signal $\boldsymbol{u} = (u_0, ..., u_{2^N-1})^\top$ and its Gaussian pyramid $\{\boldsymbol{v}^N, ..., \boldsymbol{v}^0\}$
- Define an *interpolation operator (prolongation operator)* from level $k-1$ to level k as multiplication with the $2^k \times 2^{k-1}$ matrix

$$\boldsymbol{P}_{k-1}^{k} := \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \hdotsfor{7} \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}.$$

- The *Laplacian pyramid (Laplace–Pyramide)* $\{\boldsymbol{w}^N, ..., \boldsymbol{w}^0\}$ of $\boldsymbol{u}$ is computed as

$$\begin{aligned} \boldsymbol{w}^k &:= \boldsymbol{v}^k - \boldsymbol{P}_{k-1}^{k} \boldsymbol{v}^{k-1} \qquad (k = N, ..., 1), \\ \boldsymbol{w}^0 &:= \boldsymbol{v}^0. \end{aligned}$$

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 27 28 29 30 31 32 33 34 35 36

Remarks

- Our restriction and interpolation operators fit together nicely:
  $\boldsymbol{u}^k$ and its smoothed variant $\boldsymbol{P}_{k-1}^{k} \boldsymbol{R}_{k}^{k-1} \boldsymbol{u}^k$ have the same average grey value.
- Thus, each Laplacian pyramid level has average grey value 0, except for $\boldsymbol{w}^0 = \boldsymbol{v}^0$.
- The Laplacian pyramid allows to reconstruct the Gaussian pyramid and the original image:

$$\begin{aligned} \boldsymbol{v}^0 &:= \boldsymbol{w}^0, \\ \boldsymbol{v}^k &:= \boldsymbol{w}^k + \boldsymbol{P}_{k-1}^{k} \boldsymbol{v}^{k-1} \qquad (k = 1, ..., N), \\ \boldsymbol{u} &:= \boldsymbol{v}^N. \end{aligned}$$

- The Laplacian pyramid contains more data than the original image.
  However, it has even been used for image compression:
  High-frequent components can be quantised more coarsely without visible degradations.
- To reduce aliasing, one can use more sophisticated operators.
  Example: Smooth with the binomial kernel $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$ and use linear interpolation.

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 27 28 29 30 31 32 33 34 35 36

An Application of Laplacian Pyramids: Quality Control of Nonwoven Fabrics

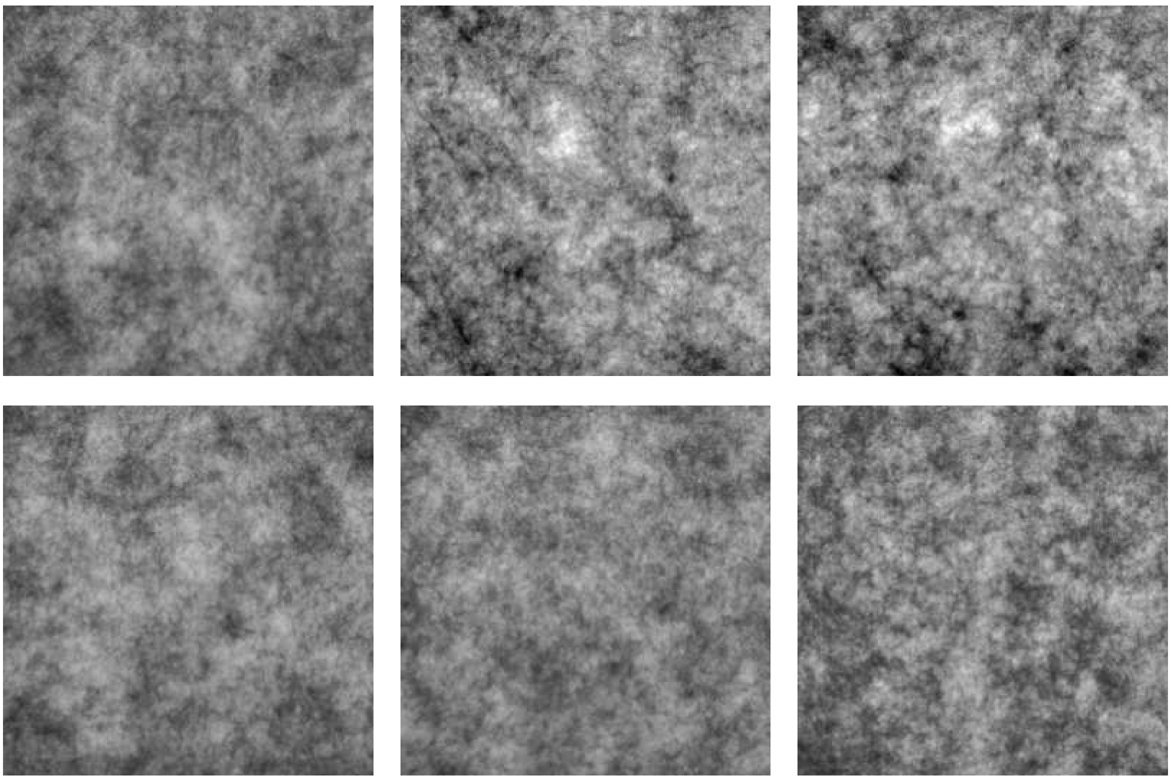
Problem

- cloudiness: essential quality parameter for nonwoven fabrics (Spinnvlies)
- company wants to automise quality control
- cloudiness is scale phenomenon: small clouds less important than larger ones

Solution

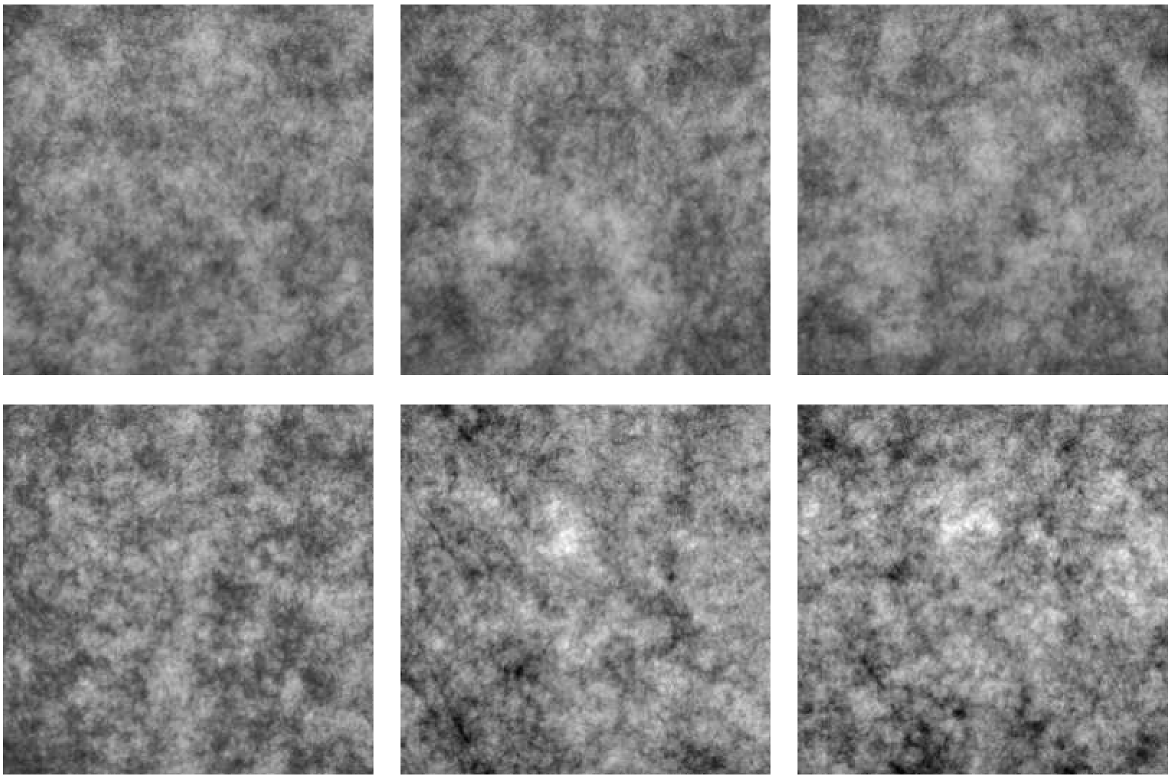
- bandpass decomposition of the fabric images using a Laplacian pyramid
- quality parameter: weighted average of grey value variances at different levels
- weights found through visual quality rankings of humans
- reliability of the system as good as a human expert
- very fast: online quality control of the entire production line on a single PC

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 27 28 29 30 31 32 33 34 35 36



Six characteristic fabric images. The goal is to order them according to their cloudiness. Author: J. Weickert.

An Application of Laplacian Pyramids (3)



From left to right and from top to bottom: Increasing cloudiness according to the automised control system. Author: J. Weickert.

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 27 28 29 30 31 32 33 34 35 36

An Application of Laplacian Pyramids (4)

Result of the fabric classification of the six images. c is the quality parameter of the system (the smaller the better), and r is the average rank (with standard deviation) given by 32 humans. This shows that the system is as reliable as these humans.

image number	#5	#1	#4	#6	#2	#3
c	48.32	51.79	52.26	66.59	85.26	100.63
r	1.50 $\pm$ 0.56	2.28 $\pm$ 0.51	2.28 $\pm$ 0.45	4.38 $\pm$ 0.54	5.03 $\pm$ 0.17	5.53 $\pm$ 0.50

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 27 28 29 30 31 32 33 34 35 36

Summary

- The discrete cosine transform (DCT) is real-valued.
  It creates signal extensions by mirroring.
- Its rapid decay of high frequency coefficients is useful for image compression.
- Pyramids are image representations in the spatial domain.
- The Gaussian pyramid acts like a lowpass filter.
  It can speed up algorithms by means of course-to-fine strategies.
- The Laplacian pyramid acts like a bandpass filter.
  It gives a frequency decomposition in the spatial domain.
- Pyramid decompositions have linear complexity.
- Pyramids are redundant: They require more disk space than the original image.
- Pyramids are not invariant under translations.

References

- N. Ahmet, T. Natarajan, K. R. Rao: Discrete cosine transform. *IEEE Transactions on Computers*, Vol. 23, pp. 90–93, Jan. 1974.
  (made the DCT popular)
- P. J. Burt, E. H. Adelson: The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, Vol. 31, No. 4, 532–540, 1983.
  (`http://persci.mit.edu/pub_pdfs/pyramid83.pdf`)
  (an early paper on image pyramids)
- E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, J. M. Ogden: Pyramid methods in image processing. *RCA Engineer*, Vol. 29, No, 6, 33–41, 1984.
  (`http://persci.mit.edu/pub_pdfs/RCA84.pdf`)
  (survey article with different applications)
- A. Rosenfeld: *Multiresolution Image Processing and Analysis*. Springer, Berlin, 1984.
  (monograph on pyramids and related concepts)
- W. L. Briggs, V. E. Henson, S. F. McCormick: *A Multigrid Tutorial*. Second Edition, SIAM, Philadelphia, 2000.
  (gentle introduction to multigrid methods for solving linear systems of equations)
- J. Weickert: A real-time algorithm for assessing inhomogeneities in fabrics. *Real-Time Imaging*, Vol. 5, No. 1, pp. 15–22, Febr. 1999.
  (`http://www.mia.uni-saarland.de/weickert/Papers/weickert-rti99.pdf`)
  (Laplacian pyramids for quality assessment of fabrics)

Announcement

- Next Wednesday is a public holiday (May 1).
- The Wednesday tutorial groups W1–W4 are replaced by a joint tutorial:

  Tuesday, April 30, 4:15 pm, E1.3, Lecture Hall 002.

  Attendance is optional.

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 27 28 29 30 31 32 33 34 35 36

Assignment C3 – Classroom Work

Problem 1 (Interpretation of the Fourier Spectrum)

In this assignment you can discover certain features of the DFT. On the next page you see three images and their logarithmically rescaled Fourier spectrum obtained with the DFT. The lowest frequencies have been shifted towards the centre of the image.

(a) Sinusoidal wave:

- Why do you observe a three-point spectrum and why is it located in this way?
- What do the intensities of the points in the spectrum stand for?
  What happens to the original image if you change them?
- What does the distance between the points in the Fourier spectrum correspond to?
- What happens to the Fourier spectrum if the sinusoidal wave is shifted in space?
- What happens if you rotate the original image? Do you expect boundary artifacts?

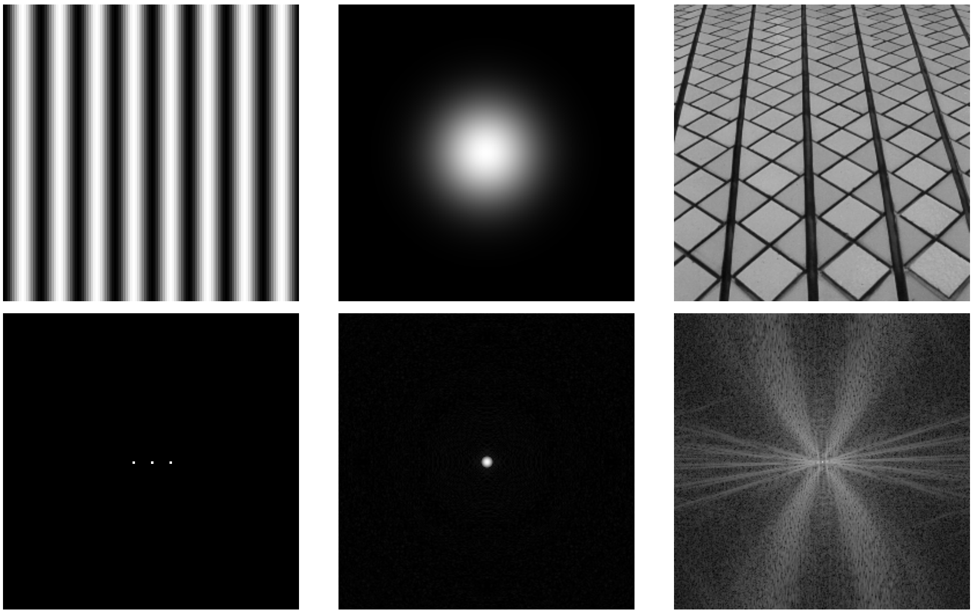
(b) Gaussian:

- What happens if we reduce the variance in the spatial domain?

(c) Downsampled tile image:

- Why do you obtain lines in your Fourier spectrum?
- Can you find aliasing artifacts in the spectrum?
- What happens in the spatial domain, if the symmetry in the Fourier spectrum is destroyed?

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 27 28 29 30 31 32 33 34 35 36



Top left: Sinusoidal wave. **Top middle:** Gaussian. **Top right:** Downsampled tile image. **Bottom row:** Respective Fourier spectra.

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 27 28 29 30 31 32 33 34 35 36

Assignment H3 – Homework

Problem 1 (Discrete Fourier Transform) (6 points)

The 1-D discrete Fourier transform expresses a signal of length M in terms of the M complex-valued vectors

$$\boldsymbol{b}_p \;:=\; \frac{1}{\sqrt{M}}\left(\exp\left(\frac{i2\pi p0}{M}\right), \exp\left(\frac{i2\pi p1}{M}\right), \ldots, \exp\left(\frac{i2\pi p(M-1)}{M}\right)\right)^{\top}$$

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

Show that these vectors are orthonormal w.r.t. the Hermitian inner product, i.e.

$$\langle \boldsymbol{b}_p, \boldsymbol{b}_q \rangle = \begin{cases} 0 & (p \neq q), \\ 1 & (p = q). \end{cases}$$

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 27 28 29 30 31 32 33 34 35 36

Assignment H3 (2)

Problem 2 (Relation between DFT and DCT) (8 points)

The goal of the assignment is to show that the DCT of a discrete signal $\boldsymbol{f} = (f_m)_{m=0}^{M-1}$ of length M is related to the DFT of a shifted signal $\boldsymbol{g} = (g_m)_{m=0}^{2M-1}$ of length $2M$ which is defined as

$$g_m := \begin{cases} f_m, & (0 \leqslant m \leqslant M-1), \\ f_{2M-1-m}, & (M \leqslant m \leqslant 2M-1). \end{cases}$$

Proceed in the following way:

(a) Perform an index shift $k := m + \frac{1}{2}$ to make $\boldsymbol{g}$ symmetric w.r.t. 0 and $2M$, respectively.

(b) Compute the DFT of the shifted signal. Consider that we defined the DFT only for integer indices.

(c) Simplify the result to obtain a sum from 0 to $M-1$ which only contains entries of $\boldsymbol{f}$. You can use the substitution $n = 2M - 1 - m$.

(d) Show the assumption. You can use the identity

$$\cos(x) = \frac{\exp(ix) + \exp(-ix)}{2}$$

(The DCT removes the redundancies that are present in the DFT. Just like the DFT implements periodic boundary extensions, the DCT realises mirrored ones.)

Assignment H3 (3)

Problem 3 (Signal Pyramids) (1+1+1+1 points)

Let the following 1-D signal be given:

$$\boldsymbol{f} := (2, 6, 9, 9, 4, 8, 11, 1)^\top .$$

(a) Calculate the Gaussian pyramid of the signal.

(b) Calculate the Laplacian pyramid of the signal.

(c) Reconstruct the initial signal from the Laplacian pyramid.

(d) The Laplacian pyramid requires more pixels than the original signal. Where is the redundancy hidden?

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 27 28 29 30 31 32 33 34 35 36

Assignment H3 (4)

Problem 4 (Filtering in the Fourier Domain) (4+2 points)

Please download the archive `Ex03.tar.gz` from the webpage

`http://www.mia.uni-saarland.de/Teaching/ipcv19.shtml`

into your own directory. You can unpack them with the command `tar xvzf Ex03.tar.gz`.

In this assignment, you can use the file `DFT.c` to modify the Fourier coefficients of an image. Compile the program using the command

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

The resulting executable program DFT performs a discrete Fourier transform (DFT), modifies the coefficients according to your changes in the method `filter`, and applies an inverse DFT afterwards.

(a) The image `smoke.pgm` is a zoom into a real world image of one of the Kuwait oil fires that occured during the Gulf War. It contains a periodic horizontal artifact which looks unpleasant. Remove it by implementing a suitable filter in the Fourier domain. Explain why you choose this kind of filter. (*Hint:* Pressing the middle mouse button under `xv` reveals the pixel coordinates. The real part `ur[i][j]` and the imaginary part `ui[i][j]` of the Fourier coefficients have the index range `i=0,...,nx-1` and `j=0,...,ny-1`.)

(b) The image `fire.pgm` is another zoom into the same image. What problems do you encounter when you try to remove the line artifacts here? Explain what causes these problems.

Assignment H3 (5)

Submission

The theoretical Problems 1,2 and 3 must be sumitted in handwritten form into the mailbox of your tutorial group **before** the lecture. The mailboxes can be found in Building E2.5 on the ground floor under the stairs. For the practical Problem 4, please submit your files as follows: Rename the main directory `Ex03` to Ex03_<your_name> and use the command

```
tar czvf Ex03_<your_name>.tar.gz Ex03_<your_name>
```

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

- the source code for `DFT.c` containing your modifications,
- the Fourier spectra as well as the backtransformed images for Problem 4,
- a text file README that contains answers to the questions in Problem 4 as well as information on all people working together for this assignment.

Please make sure that only your final version of the programs and images are included. Do **not** submit any additional files, especially no executables. 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`, `t5`, `w1`, `w2`, `w3` or `w4`.

Deadline for submission: Friday, May 3, 10 am.