Lecture 12:
Image Enhancement III:
Wavelet Shrinkage, Median Filters, M-Smoothers

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

Motivation

- In Lecture 11 we have seen that linear shift invariant lowpass filters are capable of denoising images, but they destroy important information by blurring edges.
- Are there nonlinear filters that remove noise without destroying edges?

Wavelet Shrinkage

Procedure in Three Steps (Donoho / Johnstone 1992)

- *Analysis Step:*
  Represent image in a wavelet basis (Lecture 6), from which you know/hope that
  - the useful signal part is represented by a few wavelet coefficients with large magnitude
  - typical noise such as Gaussian noise is present in numerous wavelet coefficients with small magnitude
- *Shrinkage Step:*
  Eliminate noise by shrinking wavelet coefficients with small magnitude towards 0. (This is the only nonlinear step. Do not shrink the scaling coefficient !!)
- *Synthesis Step:*
  Reconstruct the image from the modified coefficients.

This procedure also resembles image compression with wavelets (Lecture 6).

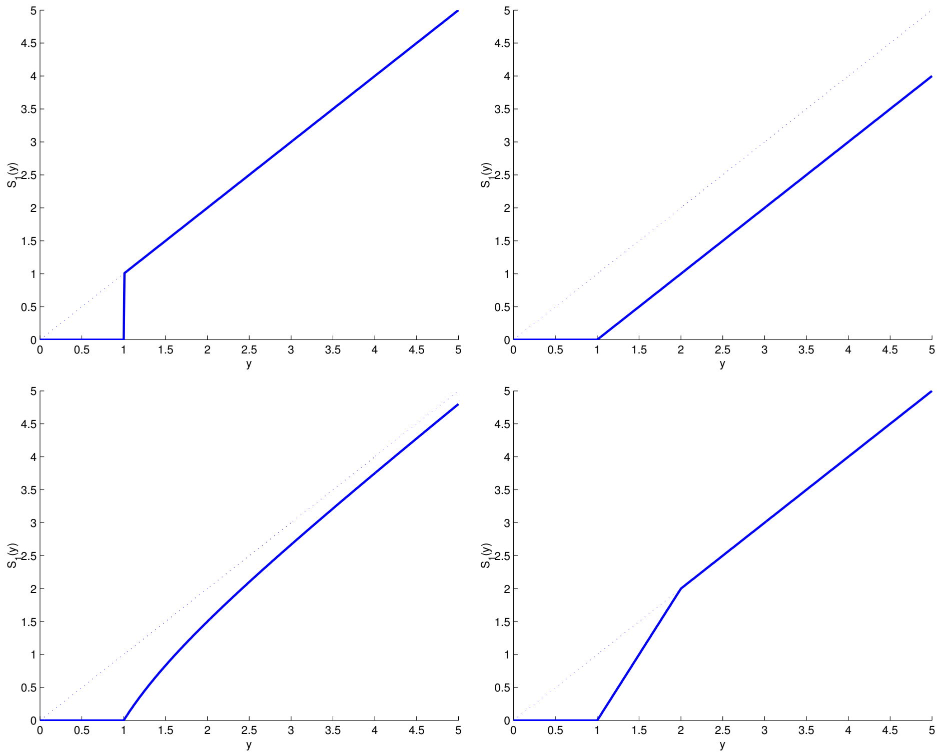
M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

Different Shrinkage Strategies

Let I_a be an indicator function, i.e. $I_a = 1$ if a is true, and $I_a = 0$ if a is false.
There are four frequent shrinkage strategies:

- *Hard Shrinkage:*
$$S_T(s) \;:=\; s\, I_{\{|s|>T\}}$$
- *Soft Shrinkage:*
$$S_T(s) \;:=\; \operatorname{sgn}(s)\,(|s|-T)\, I_{\{|s|>T\}}$$
- *Garrote Shrinkage:*
$$S_T(s) \;:=\; \left(s - \frac{T^2}{s}\right)\, I_{\{|s|>T\}}$$
- *Semisoft Shrinkage (Firm Shrinkage):*
$$S_{T_1,T_2}(s) \;=\; \begin{cases} 0 & (|s| \le T_1) \\ \operatorname{sgn}(s)\,\frac{T_2(|s|-T_1)}{T_2-T_1} & (T_1 < |s| \le T_2) \\ s & (|s| > T_2) \end{cases}$$

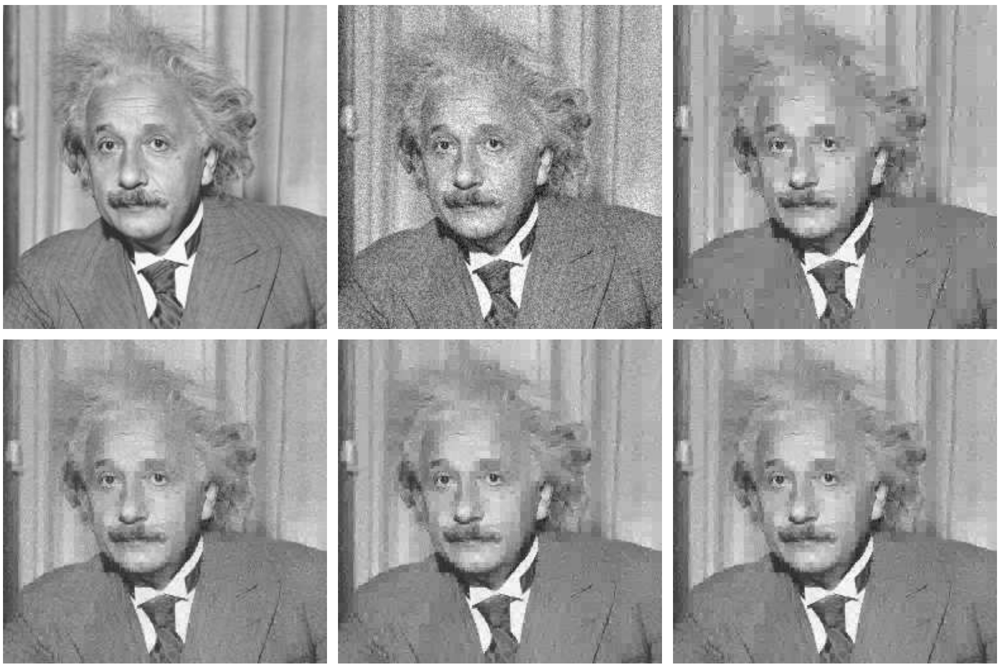
Wavelet Shrinkage (3)



Shrinkage functions. **(a) Top left:** Hard shrinkage ($T = 1$). **(b) Top right:** Soft shrinkage ($T = 1$). **(c) Bottom left:** Garrote shrinkage ($T = 1$). **(d) Bottom right:** Semisoft shrinkage ($T_1 = 1$, $T_2 = 2$). Author: P. Mrázek (2002).

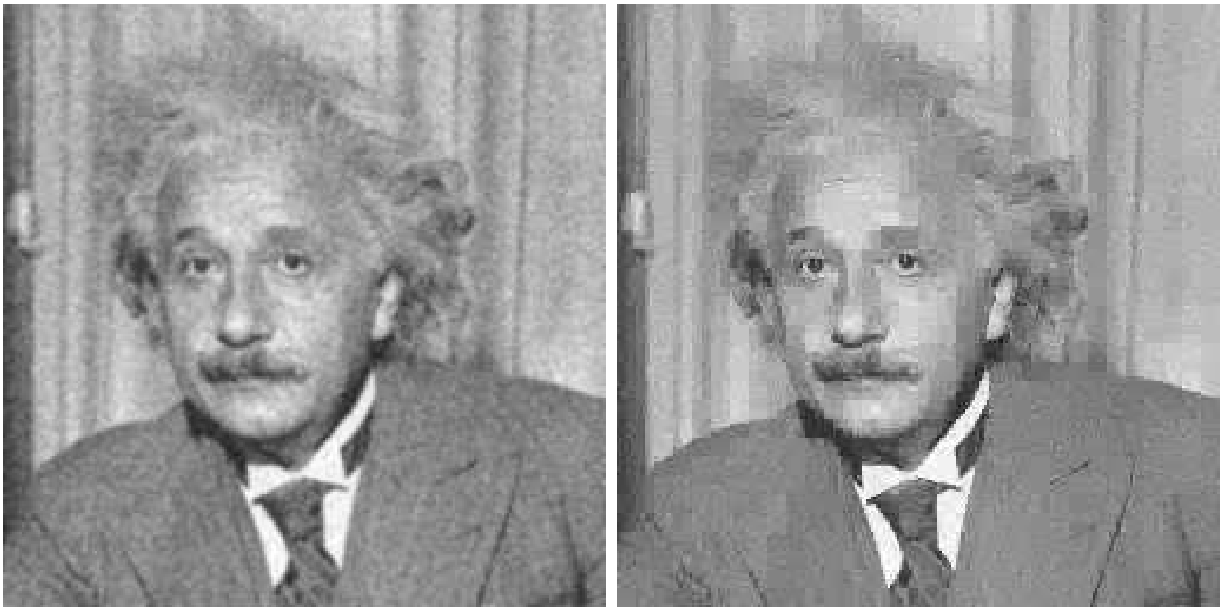
Wavelet Shrinkage (4)

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



(a) Top left: Original image, 256×256 pixels. **(b) Top middle:** With additive Gaussian noise with $\sigma = 17.3$. **(c) Top right:** Hard Haar wavelet shrinkage of (b) with $T = 50$. **(d) Bottom left:** Soft wavelet shrinkage with $T = 18$. **(e) Bottom middle:** Garrote shrinkage with $T = 29$. **(f) Bottom right:** Semisoft shrinkage with $T_1 = 24$ and $T_2 = 135$. All shrinkage parameters are chosen such that the SNR is maximised. Author: J. Weickert (2002).

Wavelet Shrinkage (5)



Comparison between a linear filter and wavelet shrinkage. **(a) Left:** Optimal denoising by Gaussian convolution ($\sigma = 0.75$). Edges are blurred. **(b) Right:** Optimal denoising with semisoft Haar wavelet shrinkage ($T_1 = 24$ und $T_2 = 135$). Edges are better preserved. Author: J. Weickert (2002).

Wavelet Shrinkage (6)

Why Does Wavelet Shrinkage Handle Edges Better Than Linear Filtering ?

Linear Shift Invariant Filtering

- Each Fourier coefficient has a *global* impact on the entire image.
- Filtering comes down to elimination or attentuation of certain Fourier coefficients.
- Thus, an LSI filter cannot be tuned to treat local structures such as edges differently than noise somewhere else.

Wavelet Shrinkage

- The impact of a single wavelet coefficient is *local*.
- Edges create wavelet coefficients with large magnitude.
- They are not (in case of hard or semisoft shrinkage) or not significantly (in case of soft or garrote shrinkage) influenced under shrinkage.

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

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

Further Properties of Wavelet Shrinkage

- It does not affect the average grey value, since wavelets have mean 0.
- It may happen that the maximum grey value increases and the minimum grey value decreases! In particular near edges, over- and undershoots appear frequently *(pseudo-Gibbs artifacts)*.
- Usual wavelet shrinkage is not shift invariant.
- Averaging over all possible translations creates shift-invariant wavelet shrinkage *(cycle spinning, algorithm a trous)*.
  Such methods do suffer less from pseudo-Gibbs artifacts. In contrast to usual wavelet shrinkage, shift invariant shrinkage can be improved by iterating.
- Usual wavelet shrinkage is also not rotationally invariant.
- For Gaussian noise with variance σ^2, there are proposals for an optimal choice of the shrinkage parameter T in soft wavelet shrinkage. For 1-D signals with N pixels:

$$T_{\text{opt}} := \sigma\sqrt{2\ln(N)}.$$

- Different shrinkage parameters at different levels may give further improvements.

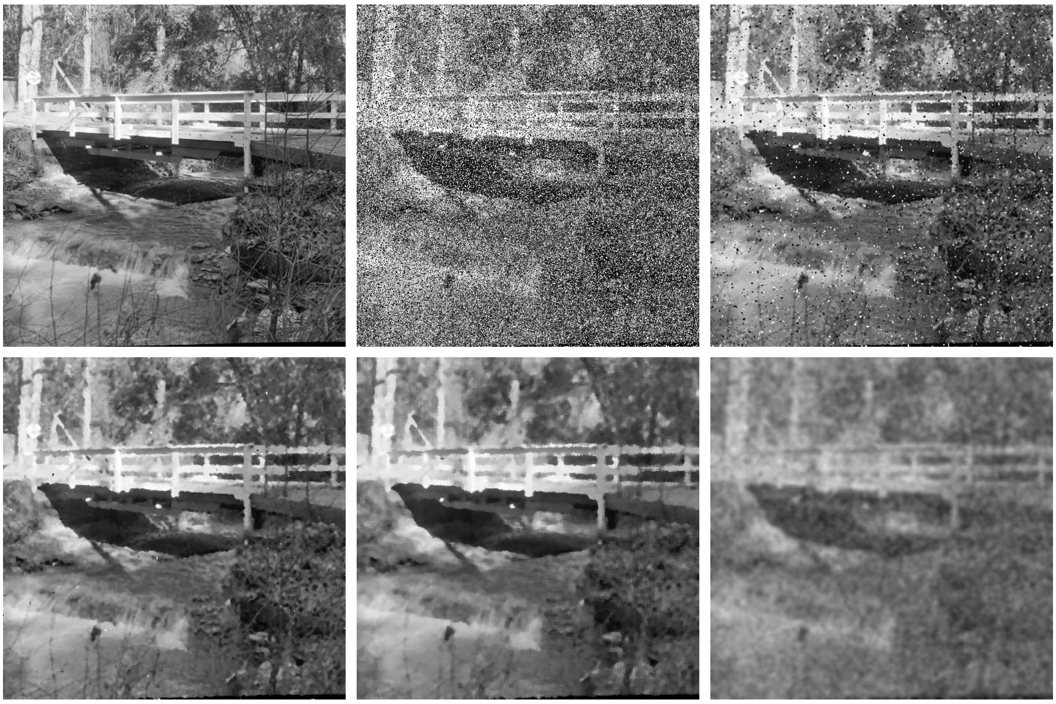
M I A

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

Median Filters (Medianfilter)

- Goal: Remove noise that creates impulse-like outliers
  (e.g. salt-and-pepper noise, cf. Lecture 2).
- Consider all grey values within a $(2m+1) \times (2m+1)$ mask.
- Create an ordering with increasing grey value.
  If the same grey value appear several times, it is also counted several times.
- As the filtered pixel, choose the *median* of this set, i.e. the value in the middle of this ordering (not the mean).
- Median filters belong to the class of *rank order filters (Rangordnungsfilter)*.

Median Filters (2)

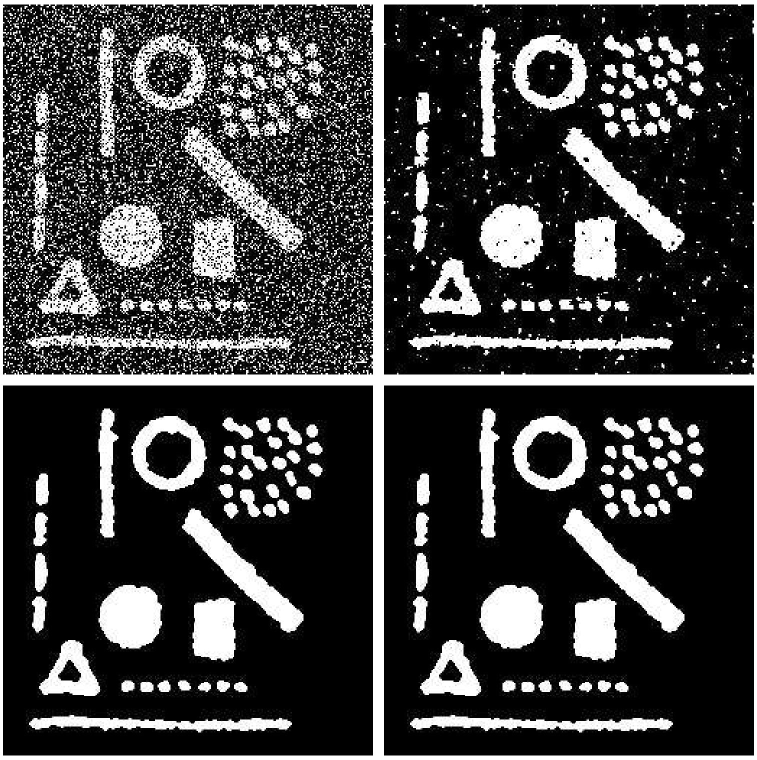


(a) Top left: Original image, 512×512 pixels. **(b) Top middle:** With 40 % salt-and-pepper noise. **(c) Top right:** Median filtering of (b) with a 3×3 mask, $\mathrm{SNR} = 5.30$ dB. **(d) Bottom left:** 5×5 median, $\mathrm{SNR} = 8.83$ dB. **(e) Bottom middle:** 7×7 median, $\mathrm{SNR} = 8.18$ dB. **(f) Bottom right:** Gaussian smoothing, $\sigma = 3$, $\mathrm{SNR} = -0.41$ dB. Author: J. Weickert (2005).

Median Filters (3)

Properties of Median Filters

- Median filters are invariant under gamma correction, histogram equalisation and other monotone increasing rescalings of the image greyscale (why ?).
- Filters with this invariance are called *morphological filters* (see next lecture).
- appropriate for removing impulse noise:
  A single outlier does not affect the median of a set.
- preserves (straight or almost straight) edges better than linear filters
- leads to roundings of corners (why ?)
- Usually, it stabilises after a few iterations to a so-called *root signal*.
- Example of an unstable exeption:
  median filtering of the infinite 1-D signal $(..., 0, 1, 0, 1, 0, 1, ...)$ oscillates
- Median filters are not separable!
- Sorting can be rather time consuming.
- Remedy: By moving the $(2m+1) \times (2m+1)$ mask by one pixel, one has to exchange only $(2m+1)$ grey values in the sorted list.



(a) Top left: Test image with 40 % salt-and-pepper noise. **(b) Top right:** Filtered with a 3×3 median, 1 iteration. **(c) Bottom left:** 16 iterations. **(d) Bottom right:** Using 256 iteration instead does not improve the image any further. Author: J. Weickert (2000).

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

M-Smoothers (M-Glätter)

- formalism for decribing a number of local filters within an optimisation framework
- Given: discrete image $\mathbf{f} = (f_i)$.
- Wanted: filtered image $\mathbf{u} = (u_i)$.
- The filtered grey value u_i minimises a kind of distance to its original grey values f_j within the neighbourhood $B(i)$ of pixel i.
- More precisely: Replace the grey value f_i by its *local M-estimate*

$$\begin{aligned} u_i &:= \operatorname{argmin} \Phi_i(y) \\ \Phi_i(y) &:= \sum_{j \in B(i)} \varphi(|f_j - y|) \end{aligned}$$

  with a monotone increasing (penalising) function φ.
- The choice of the penaliser φ determines the influence that is assigned to strong deviations. In this way it is possible to construct edge-preserving and even edge-enhancing filters.

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

An Important Class of Penalisers: $\varphi(s) := s^p \qquad (0 < p < \infty)$

The Case $p \to \infty$: Midrange Filtering

- One can show that, for $p \to \infty$, the minimiser of the p-distances

$$\Phi(y) = \sum_{j \in B(i)} |f_j - y|^p$$

is given by the *midrange*

$$u_i = \frac{\max\limits_{j \in B(i)} f_j + \min\limits_{j \in B(i)} f_j}{2}$$

- Midrange filtering is very sensitive to outliers.
- It blurs edges by incorporating the most extreme grey values from both sides of the edge in the averaging process.

The Case $p = 2$: Linear Box Filtering

- The filtered grey value u_i minimises the quadratic distance

$$\Phi(y) = \sum_{j \in B(i)} |f_j - y|^2.$$

- Assignment: Show that u_i is given by the *arithmetic mean*

$$u_i = \frac{1}{|B(i)|} \sum_{j \in B(i)} f_j$$

where $|B(i)|$ denotes the number of elements of the set $B(i)$.

- Thus, we have a (linear) *box filter* with mask $B(i)$.
- It is sensitive to outliers and blurs edges.

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

The Case $p = 1$: Median Filtering

- The filtered grey value u_i minimises the distance

$$\Phi(y) = \sum_{j \in B(i)} |f_j - y|.$$

- Assignment: This leads to the median within $B(i)$.
- *Median filters are specific M-smoothers!*
- We have seen that median filters are robust against outliers and preserve edges.

- **The Case $p \to 0$: Mode Filtering**
  - For $p < 1$ multiple local minima may pop up:
    The optimisation problem is no longer convex.
  - One can show that in the limit $p \to 0$ all maxima *(modes, Modalwerte)* of the local histogram within $B(i)$ are minimisers of the distance function.
  - Filters that replace a grey value by the most frequent grey value within a neighbourhood are called *mode filters (Modalfilter)*.
  - Mode filters can be robust under noise (but not for salt-and-pepper noise) and may even enhance edges.
  - Their correct implementation is highly nontrivial.

Summary

- Wavelet shrinkage eliminates wavelet coefficients with small magnitude.
- 4 most important shrinkage functions: hard, soft, garrote, semisoft.
- The locality of wavelets allows the preservation of edges.
- The average grey value is preserved as well.
- Median filters replace the grey value by its median within a $(2m+1) \times (2m+1)$ mask.
- They are well-suited for edge-preserving removal of impulse noise.
- M-smoothers consitute a general filter class, in which a number of important local filters are minimising specific distance measures.
- Special cases include midrange filtering, box filtering, median filtering and mode filtering.

Literature

- W. Bäni: *Wavelets*. Oldenbourg, München, 2002.
  (fairly simple introduction to wavelet concepts; in German)
- S. Mallat: *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, Second Edition, 1999.
  (wavelet bible)
- I. K. Folder, C. Kamath: Denoising through wavelet shrinkage: an empirical study. *Journal of Electronic Imaging*, Vol. 12, No. 1, 151–160, Jan 2003.
  (`http://citeseer.ist.psu.edu/590032.html`)
  (evaluates different shrinkage strategies)
- R. Fisher, S. Perkins, A. Walker, E. Wolfart: *Median Filter*.
  (`www.dai.ed.ac.uk/HIPR2/median.htm`).
  (useful internet resource on median filtering)
- J. Barral Souto: *El modo y otras medias, casos particulares de una misma expresión matemática*. Cuadernos de Trabajo No. 3, Instituto de Biometria, Universidad Nacional de Buenos Aires, Argentina, 1938.
  (derives midrange, median, mean, mode as minimisers; in Spanish, but well-written)
- P. L. Torroba, N. L. Cap, H. J. Rabal, W. D. Furlan: Fractional order mean in image processing. *Optical Engineering*, Vol. 33, No. 2, pp. 528–534, 1994.
  (investigates M-smoothers for different p)

Assignment P3 – Programming Work

Please download the required files from the webpage

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

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

Problem 1 (Affine Rescaling) (4 + 2 points)

The programme `pointtransTemplate.c` contains the subroutine `rescale`. It is supposed to perform an affine greyscale transformation such that the rescaled image has the greyscale range $[a, b]$.
As for all point transformations, this can be realised by specifying the entries of an integer 1-D mapping array `g[i]` that assigns to each possible input grey value `i` $\in [0, 255]$ its new output grey value `g[i]`.

(a) Supplement the missing code and compile your programme with

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

Note that the image `u[i][j]` is defined in the index range `i=1,...,nx` and `j=1,...,ny`.

(b) Test the routine with the image `machine.pgm`. In order to determine if the rescaling works correctly use at least one setting with $a \neq 0$.

Problem 2 (Gamma Correction) (4 + 2 points)

Gamma correction is an important nonlinear point operation. For an image with greyscale range $[0, 255]$ it has the structure

$$\phi(f(x,y)) := 255 \left(\frac{f(x,y)}{255} \right)^{1/\gamma} .$$

(a) Implement the method in the subroutine `gamma_correct` of programme `pointtransTemplate.c`.

(b) Validate it with the image `asbest.pgm`. What values for γ give reasonable results ?

Problem 3 (Histogram Equalisation) (6 + 2 points)

The last point transformation that shall be addressed in this programming exercise is the equalisation of an image histogram.

(a) Complete the subroutine `hist_equal` in the programme `pointtransTemplate.c` such that it performs this task in accordance with the algorithm presented in the lecture.

(b) Test your implementation with the image is `office.pgm`.

Assignment P3 (3)

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 `Ex03` to Ex03_<your_name> and use the command

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

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

- the source code for `pointtransTemplate.c` with the subroutines for the problems 1–3
- the corresponding test images with applied point operations
- a text file README that contains the answer to the question in problem 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, December 11, 10 am (before the lecture)