Image Processing and Computer Vision (IPCV)

Prof. J. Weickert
Mathematical Image Analysis Group

Winter term 13/14
Saarland University

Example Solutions for Homework Assignment 10 (H10)

Problem 1 (Discrete Deconvolution)

(a)+(b) As a necessary condition for a minimiser of $G(\boldsymbol{u})$ the first partial derivatives w.r.t. $u_1, \ldots, u_N$ must vanish:

$$\begin{aligned}
0 = \frac{\partial G}{\partial u_1} = \; & \frac{1}{2}\Bigg(\; 2\big(\tfrac{1}{4}u_1 + \tfrac{1}{2}u_1 + \tfrac{1}{4}u_2 - f_1\big)\big(\tfrac{1}{4} + \tfrac{1}{2}\big) \\
& \qquad + 2\big(\tfrac{1}{4}u_1 + \tfrac{1}{2}u_2 + \tfrac{1}{4}u_3 - f_2\big)\tfrac{1}{4}\Bigg) \\
& + \frac{\alpha}{2}\Bigg(\; 2\big(u_2 - u_1\big)(-1)\Bigg) \\
= \; & \left(\frac{5}{8} + \alpha\right)u_1 + \left(\frac{5}{16} - \alpha\right)u_2 + \frac{1}{16}u_3 - \left(\frac{3}{4}f_1 + \frac{1}{4}f_2\right)
\end{aligned}$$

$$
\begin{aligned}
0 = \quad \frac{\partial G}{\partial u_2} = \quad & \frac{1}{2}\Bigg(\quad 2\big(\tfrac{1}{4}u_1 + \tfrac{1}{2}u_1 + \tfrac{1}{4}u_2 - f_1\big)\tfrac{1}{4} \\
& \qquad +2\big(\tfrac{1}{4}u_1 + \tfrac{1}{2}u_2 + \tfrac{1}{4}u_3 - f_2\big)\tfrac{1}{2}\Bigg) \\
& \qquad +2\big(\tfrac{1}{4}u_2 + \tfrac{1}{2}u_3 + \tfrac{1}{4}u_4 - f_3\big)\tfrac{1}{4}\Bigg) \\
& +\frac{\alpha}{2}\Bigg(\quad 2\big(u_2 - u_1\big)(1) \quad + \quad 2\big(u_3 - u_2\big)(-1)\Bigg) \\
= \; & \left(\frac{5}{16} - \alpha\right)u_1 + \left(\frac{3}{8} + 2\alpha\right)u_2 + \left(\frac{1}{4} - \alpha\right)u_3 + \frac{1}{16}u_4 \\
& -\left(\frac{1}{4}f_1 + \frac{1}{2}f_2 + \frac{1}{4}f_3\right)
\end{aligned}
$$

$\forall i \in \{3, \ldots N-2\}$:

$$
\begin{aligned}
0 = \quad \frac{\partial G}{\partial u_i} = \quad & \frac{1}{2}\Bigg(\quad 2\big(\tfrac{1}{4}u_i + \tfrac{1}{2}u_{i+1} + \tfrac{1}{4}u_{i+2} - f_{i+1}\big)\tfrac{1}{4} \\
& \qquad +2\big(\tfrac{1}{4}u_{i-1} + \tfrac{1}{2}u_i + \tfrac{1}{4}u_{i+1} - f_i\big)\tfrac{1}{2}\Bigg) \\
& \qquad +2\big(\tfrac{1}{4}u_{i-2} + \tfrac{1}{2}u_{i-1} + \tfrac{1}{4}u_i - f_{i-1}\big)\tfrac{1}{4}\Bigg) \\
& +\frac{\alpha}{2}\Bigg(\quad 2\big(u_i - u_{i-1}\big)(1) \quad + \quad 2\big(u_{i+1} - u_i\big)(-1)\Bigg) \\
= \; & \frac{1}{16}u_{i-2} + \left(\frac{1}{4} - \alpha\right)u_{i-1} + \left(\frac{3}{8} + 2\alpha\right)u_i + \left(\frac{1}{4} - \alpha\right)u_{i+1} + \frac{1}{16}u_{i+2} \\
& -\left(\frac{1}{4}f_{i-1} + \frac{1}{2}f_i + \frac{1}{4}f_{i+1}\right)
\end{aligned}
$$

$$
\begin{aligned}
0 = \quad \frac{\partial G}{\partial u_{N-1}} = \quad & \frac{1}{2}\Bigg(\quad 2\big(\tfrac{1}{4}u_{N-1} + \tfrac{1}{2}u_N + \tfrac{1}{4}u_N - f_N\big)\tfrac{1}{4} \\
& \qquad +2\big(\tfrac{1}{4}u_{N-2} + \tfrac{1}{2}u_{N-1} + \tfrac{1}{4}u_N - f_{N-1}\big)\tfrac{1}{2}\Bigg) \\
& \qquad +2\big(\tfrac{1}{4}u_{N-3} + \tfrac{1}{2}u_{N-2} + \tfrac{1}{4}u_{N-1} - f_{N-2}\big)\tfrac{1}{4}\Bigg) \\
& +\frac{\alpha}{2}\Bigg(\quad 2\big(u_N - u_{N-1}\big)(-1) \quad + \quad 2\big(u_{N-1} - u_{N-2}\big)(1)\Bigg) \\
= \; & \left(\frac{5}{16} - \alpha\right)u_N + \left(\frac{3}{8} + 2\alpha\right)u_{N-1} + \left(\frac{1}{4} - \alpha\right)u_{N-2} + \frac{1}{16}u_{N-3} \\
& -\left(\frac{1}{4}f_{N-2} + \frac{1}{2}f_{N-1} + \frac{1}{4}f_N\right)
\end{aligned}
$$

$$
\begin{aligned}
0 = \frac{\partial G}{\partial u_N} = \quad & \frac{1}{2}\Bigg(\quad 2\big(\tfrac{1}{4}u_{N-1} + \tfrac{1}{2}u_N + \tfrac{1}{4}u_N - f_N\big)\big(\tfrac{1}{2} + \tfrac{1}{4}\big) \\
& \qquad + 2\big(\tfrac{1}{4}u_{N-2} + \tfrac{1}{2}u_{N-1} + \tfrac{1}{4}u_N - f_{N-1}\big)\tfrac{1}{4}\Bigg) \\
& + \frac{\alpha}{2}\Bigg(\quad 2\big(u_N - u_{N-1}\big)(1)\Bigg) \\
= \; & \left(\frac{5}{8} + \alpha\right)u_N + \left(\frac{5}{16} - \alpha\right)u_{N-1} + \frac{1}{16}u_{N-2} - \left(\frac{1}{4}f_{N-1} + \frac{3}{4}f_N\right)
\end{aligned}
$$

The linear system of equations with the unknowns $u_1, \ldots, u_N$ is given by

$$
\begin{pmatrix}
\frac{5}{8}+\alpha & \frac{5}{16}-\alpha & \frac{1}{16} & & & & & & \\
\frac{5}{16}-\alpha & \frac{3}{8}+2\alpha & \frac{1}{4}-\alpha & \frac{1}{16} & & & & & \\
\frac{1}{16} & \frac{1}{4}-\alpha & \frac{3}{8}+2\alpha & \frac{1}{4}-\alpha & \frac{1}{16} & & & & \\
& \ddots & \ddots & \ddots & \ddots & \ddots & & & \\
& & \frac{1}{16} & \frac{1}{4}-\alpha & \frac{3}{8}+2\alpha & \frac{1}{4}-\alpha & \frac{1}{16} & & \\
& & & \ddots & \ddots & \ddots & \ddots & \ddots & \\
& & & & \frac{1}{16} & \frac{1}{4}-\alpha & \frac{3}{8}+2\alpha & \frac{1}{4}-\alpha & \frac{1}{16} \\
& & & & & \frac{1}{16} & \frac{1}{4}-\alpha & \frac{3}{8}+2\alpha & \frac{5}{16}-\alpha \\
& & & & & & \frac{1}{16} & \frac{5}{16}-\alpha & \frac{5}{8}+\alpha
\end{pmatrix}
\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_i \\ \vdots \\ u_{N-2} \\ u_{N-1} \\ u_N \end{pmatrix}
=
\begin{pmatrix}
\frac{3}{4}f_1 + \frac{1}{4}f_2 \\
\frac{1}{4}f_1 + \frac{1}{2}f_2 + \frac{1}{4}f_3 \\
\frac{1}{4}f_2 + \frac{1}{2}f_3 + \frac{1}{4}f_4 \\
\vdots \\
\frac{1}{4}f_{i-1} + \frac{1}{2}f_i + \frac{1}{4}f_{i+1} \\
\vdots \\
\frac{1}{4}f_{N-3} + \frac{1}{2}f_{N-2} + \frac{1}{4}f_{N-1} \\
\frac{1}{4}f_{N-2} + \frac{1}{2}f_{N-1} + \frac{1}{4}f_N \\
\frac{1}{4}f_{N-1} + \frac{3}{4}f_N
\end{pmatrix}
$$

where empty entries are 0.

At most, there are 5 nonvanishing entries in the rows of the system matrix.

(c) A general blurring kernel of size $n > 1$ incorporates at most $n-1$ pixels to the left and to the right side, considering the partial derivatives of the energy functional. Hence, including the central pixel, we would expect $2(n-1)+1 = 2n-1$ nonvanishing entries in the system matrix.

So we see that depending on the convolution kernel the system matrices are in the case of variational deconvolution not that sparse anymore. Such systems can be solved efficiently in the Fourier domain if the kernel h is not space-variant.

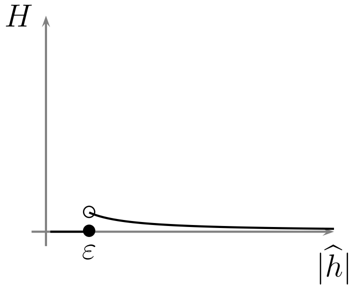
Problem 2 (Deconvolution Methods)

(a) Just as in Problem 2 of C10, we sketch H in dependence of $|\hat{h}|$. For a more detailed discussion of this choice, please consult the example solution for C10.

- *Pseudoinverse filtering* is defined as

$$\widehat{u} = \begin{cases} \frac{1}{\widehat{h}}\widehat{f}, & \text{if } |\widehat{h}| > \varepsilon \\ 0, & \text{else.} \end{cases}$$

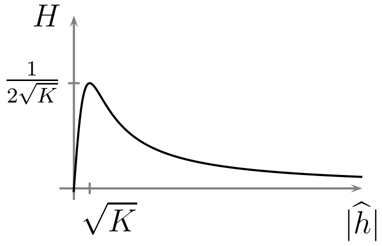
The following figure depicts the function $H(|\widehat{h}|) := \begin{cases} \frac{1}{|\widehat{h}|}, & \text{if } |\widehat{h}| > \varepsilon \\ 0, & \text{else.} \end{cases}$



- *Wiener filtering* is defined as

$$\widehat{u} = \underbrace{\frac{1}{\widehat{h}} \frac{|\widehat{h}|^2}{|\widehat{h}|^2 + K}}_{=:\widehat{H}(\widehat{h})} \widehat{f}.$$

The following Figure depicts the function $\widehat{H}(|\widehat{h}|) = \frac{1}{|\widehat{h}|}\frac{|\widehat{h}|^2}{|\widehat{h}|^2+K}$.



(b) *Pseudoinverse filtering* acts in a similar way as a bandpass filter: For low frequencies $|\widehat{h}|$ is large, and so H corresponds there the theoretically optimal filter result, namely inverse filtering. For heigh frequencies $|\widehat{h}|$ is small (almost zero). There, depending on the choice of ε, pseudoinverse filtering sets H manually to zero. Thus the highpass effect as given in inverse filtering is avoided and tiny high-frequent noise is not amplified.

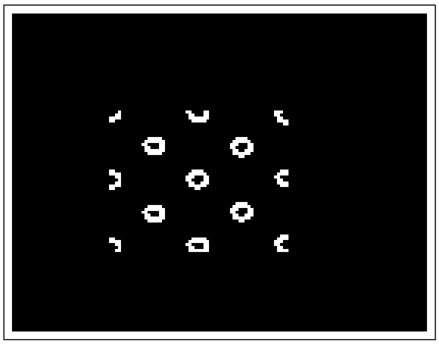
We see, the parameter ε determines which of the high frequencies are considered to be noise and are thus set to zero to avoid an amplification. Consequently, ε should be chosen depending on the noise expected in the given data: The more noise expected the larger ε should be chosen.

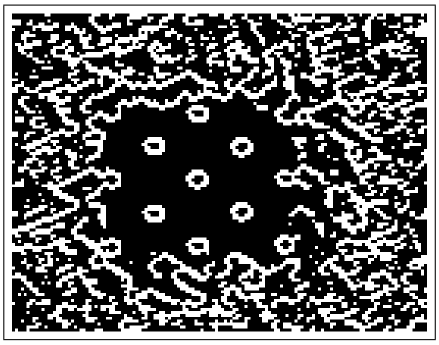
Finally let us consider *Wiener filtering.* If h is a lowpass filter, we know that high frequencies are attenuated and therefore, the values for $|\widehat{h}|$ are very small. As one can see in the sketch for Wiener filtering, extremely high frequencies (very small values for $|\widehat{h}|$) are also attenuated by H as H converges towards 0 for $|\widehat{h}| \to 0$. In contrast, for $|\widehat{h}| \to \infty$ (that is low frequencies which lead to high values for $|\widehat{h}|$), H converges towards $\frac{1}{|\widehat{h}|}$. This is the theoretically optimal solution given by inverse filtering (see C10 P2). So low frequencies are attenuated by H as well as high frequencies. Only frequencies that lie within a certain band (determined by the value for K) pass the filter. This is the reason why, Wiener filtering acts like a bandpass filter. The advantage over pseudoinverse filtering is, that for Wiener filtering H is continuous which avoids artifacts in the filtered result.

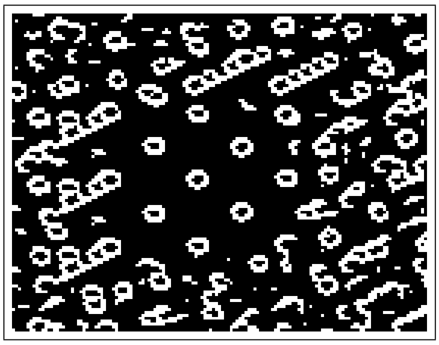
Since the parameter K, more precisely $\sqrt{K}$ corresponds to the extremum of H, the frequencies that have the value $\sqrt{K}$ after applying the filter h are amplified the most. Moreover the smaller K the higher the frequencies that are amplified. This explains why K should increase with the estimated noise variance σ^2, e.g. by choosing $K := 2\sigma^2$, as given on the slides.

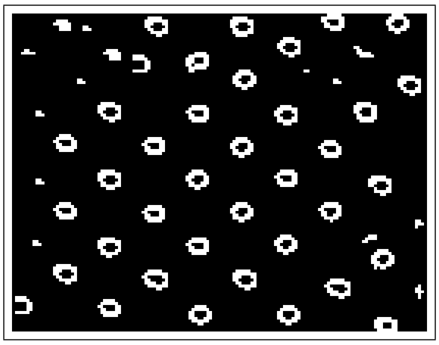
Problem 3 (Texture Inpainting)

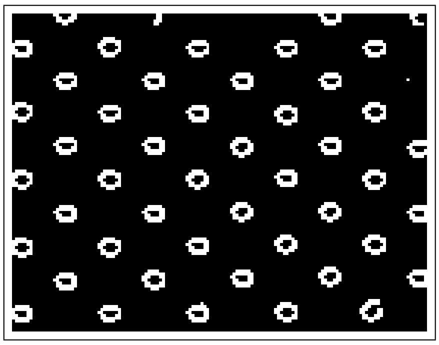
(a) The patch-size determines the amount of randomness in the newly generated texture. Small neighbourhoods tend to produce inpainting results that deviate significantly from the original texture sample. In the circle example, small choices like $m = 2$ and $m = 3$ lead to irregular line patterns, while large choices like $m = 10$ lead to verbatim copying and thereby a perfectly regular extension of the texture.

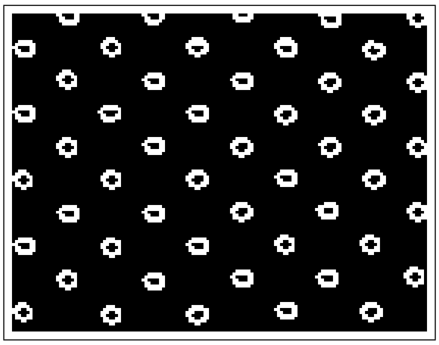

`circle.pgm`


$m = 2$


$m = 3$


$m = 6$

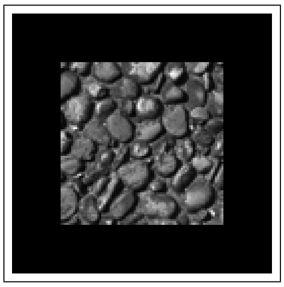

$m = 8$


$m = 10$

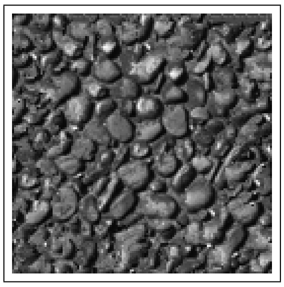
(b) The different choice of parameters depending on the image type is illustrated by the three examples from this part of the problem. The stone texture is preferably extended with a small patch size (e.g. $m = 2$), since verbatim copying does not lead to believable, natural results. Here, a certain amount of randomness is needed.

The example cyrus.pgm however looks best with large patch sizes like $m = 20$, since verbatim copying of letters is actually benifical for the authenticity of the inpainting. Since this example shows an excerpt from the ancient cyrus cylinder and is written in Akkadian cuneiform script, you would also (probably) not be able to tell the generated texture from the original image. If the meaning of symbols is not known, the results can be extremely convincing.

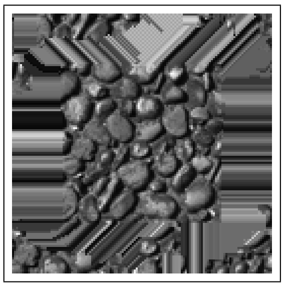
For the example *boat*, a good choice for the parameter lies somewhere inbetween the previous extreme choices (e.g. $m = 7$). This scene is much more complex than the previous examples and clearly displays the limitations of patch-based approaches. Since for some areas in the image (e.g. sky, ground), there is only limited original material, the algorithm will make odd choices. The best fitting known patches are just not a good approximation of the missing ground truth.



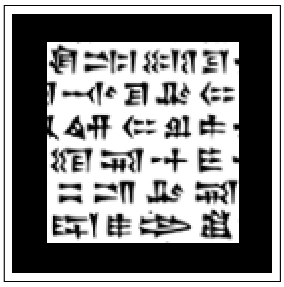
`stone.pgm`



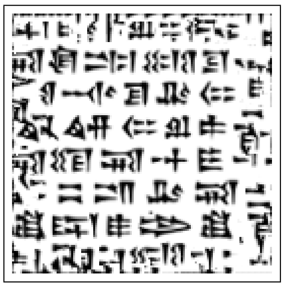
$m = 2$



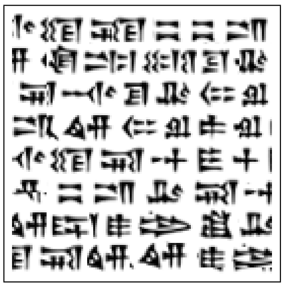
$m = 10$



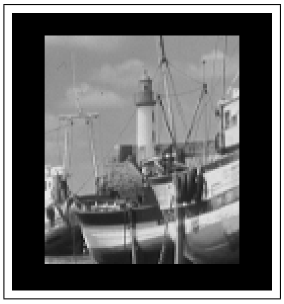
`cyrus.pgm`



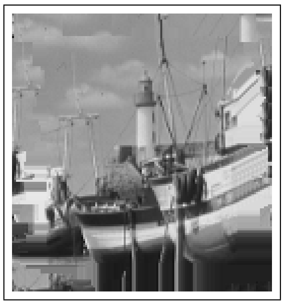
$m = 2$



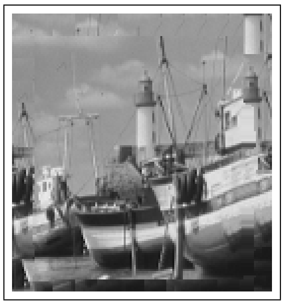
$m = 20$



`boat.pgm`



$m = 7$



$m = 10$

Problem 4 (Deconvolution with Wiener Filtering)

(a) Let f denote an image which has been degraded by convolution with the kernel h. We want to apply Wiener filtering to obtain a filtered version u by

$$\hat{u} = \left(\frac{1}{\hat{h}} \, \frac{|\hat{h}|^2}{|\hat{h}|^2 + K} \right) \hat{f} \ , \tag{1}$$

where $K > 0$ is a real positive number.

For an implementation of this formula we have to keep in mind that the symbols $\hat{f}, \hat{h}$, and $\hat{u}$ stand for Fourier coefficients, which are in general complex numbers. We use the facts

$$\frac{1}{\hat{h}} = \frac{1\,\overline{\hat{h}}}{\hat{h}\,\overline{\hat{h}}} = \frac{\overline{\hat{h}}}{|\hat{h}|^2} \quad \text{and} \quad \hat{h}\overline{\hat{h}} = |\hat{h}|^2 = \left(\mathrm{Re}(\hat{h})\right)^2 + \left(\mathrm{Im}(\hat{h})\right)^2$$

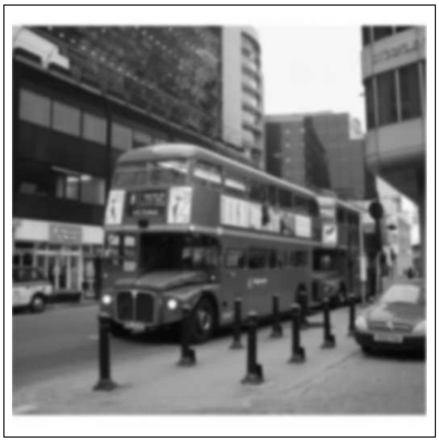
to rewrite the formula

$$\frac{1}{\hat{h}} \, \frac{|\hat{h}|^2}{|\hat{h}|^2 + K} \quad = \quad \frac{\overline{\hat{h}}}{|\hat{h}|^2 + K} \ . \tag{2}$$

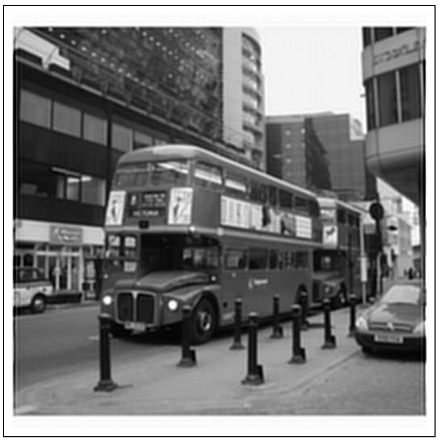
This can simplify the implementation of the complex arithmetic:

```
/*--------------------------------------------------------------*/

void filter
 (float    **ur,    /* real part of Fourier coeffs, changed */
  float    **ui,    /* imag. part of Fourier coeffs, changed */
  float    **hr,    /* real part of Fourier kernel, unchanged */
  float    **hi,    /* imag. part of Fourier kernel, unchanged */
  float    param,   /* filter parameter */
  long     nx,      /* pixel number in x direction */
  long     ny)      /* pixel number in y direction */

/* Performs Wiener Filtering in the Fourier domain. */

{
long   i, j;      /* loop variables */
float  N;         /* denominator */
float  vr,vi;     /* auxiliary variables for cplx arithm. */
/* ---- compute filtered coefficients ---- */

for (i=0; i<=nx-1; i++)
  for (j=0; j<=ny-1; j++)
    {
        /* compute the denominator */
        N = hr[i][j] * hr[i][j] + hi[i][j] * hi[i][j] + param;

        /* numerator for the real part */
        vr = hr[i][j] * ur[i][j] + hi[i][j] * ui[i][j];

        /* numerator for the imaginary part */
        vi = hr[i][j] * ui[i][j] - hi[i][j] * ur[i][j];

        ur[i][j] = vr / N;
        ui[i][j] = vi / N;
    }

return;
}
/*------------------------------------------------------------*/
```

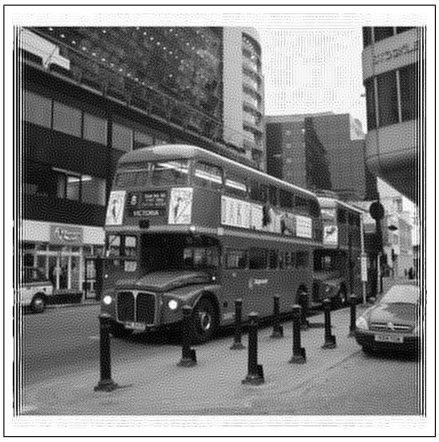
(b) We now show the influence of the parameter K on the filtering results. First we take a look at the image `bus1.pgm`. For this image, a parameter near $K = 0.01$ yields relatively good results. We see that too small parameters tend to result in artifacts near the boundary of the image as well as small high-frequent artifacts all over the image. If K tends to zero, Wiener filtering suffers from the same problems as Inverse filtering. On the other hand, choosing the parameter K too large reduces the contrast of the image. This can be seen from formula (1): A large K results in damping all Fourier coefficients, and the image becomes darker.



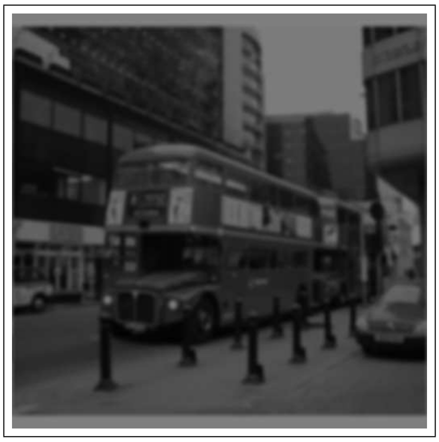
Initial image (`bus1.pgm`)



$K = 0.01$

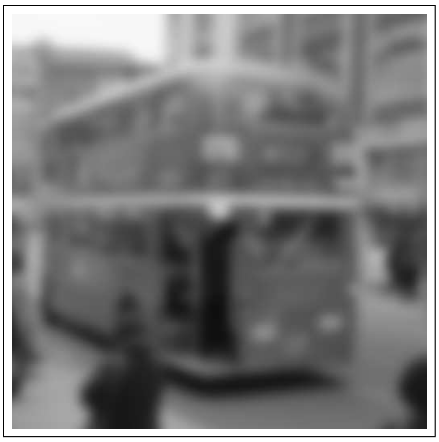


$K = 0.00001$

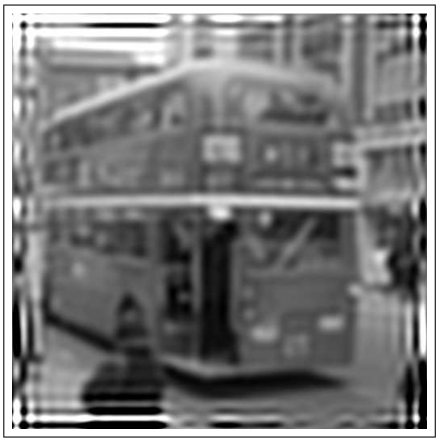


$K = 1.0$

(c) The degradations are much stronger for the second image `bus2.pgm`. Here we have to choose a smaller K to obtain a good reconstruction.

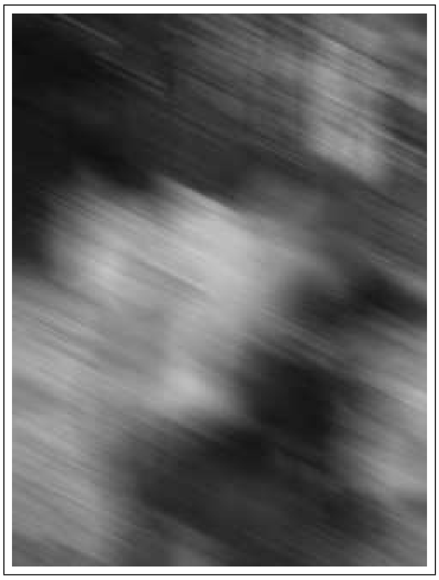


Initial image (`bus2.pgm`)

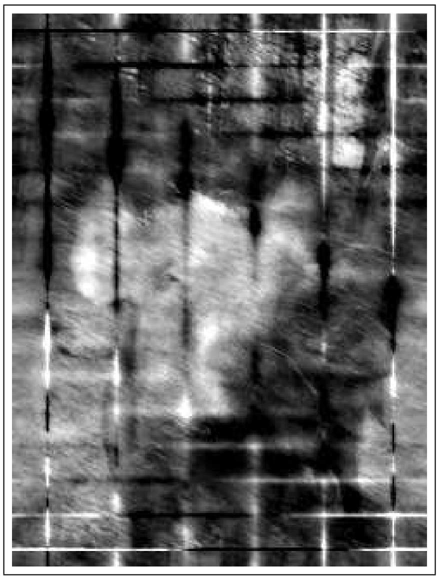


$K = 0.001$

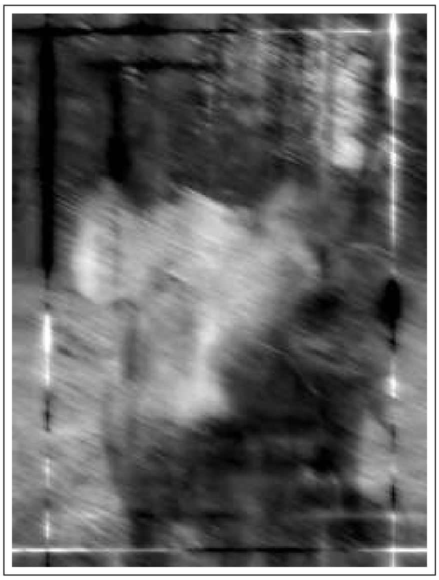
(d) The trade-off between sharpness and artifacts can be seen for the third image, `hogblur.pgm`. In this case the filtering takes much longer, since the image size is not a power of 2. Thus only the DFT and not the FFT can be used to switch between the spatial and the frequency (Fourier) domain. This huge difference in runtime compared to the previous examples makes the advantages of the complexity $N \log N$ (FFT) compared to the complexity N^2 (DFT) explicit.



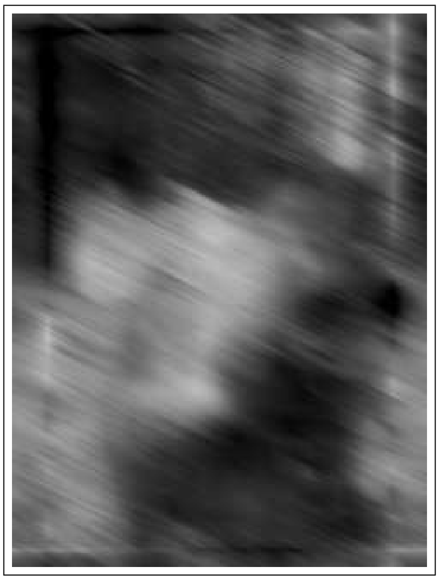
Initial image (`hogblur.pgm`)



$K = 0.001$



$K = 0.01$



$K = 0.1$