

Problem 1 (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

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

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

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

to rewrite the formula

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

This can simplify the implementation of the complex arithmetics:

```

/*-----*/

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;
}
/*-----*/

```

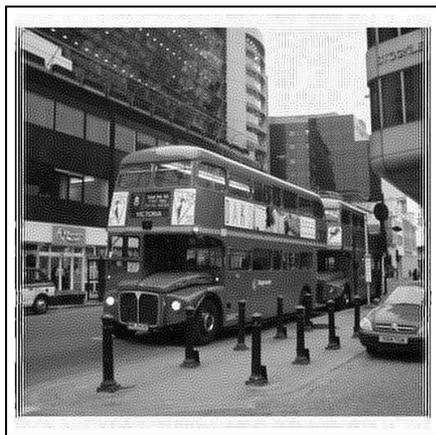
- (b) We now show the influence of the parameter K on the filtering results. First we take a look at the image `bus1.pgm`. For this image, a parameter near $K = 0.01$ yields relatively good results. We see that too small parameters tend to result in artifacts near the boundary of the image as well as small high-frequency 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`)

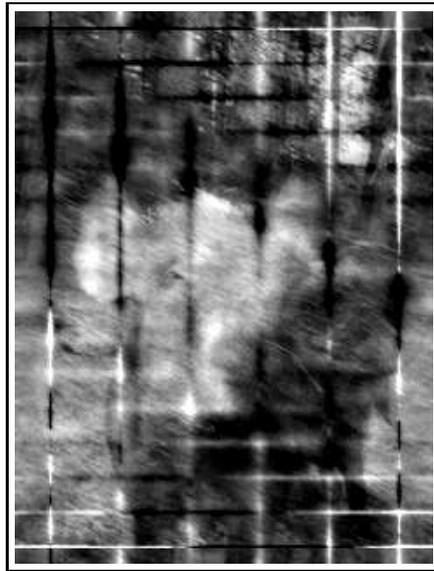


$K = 0.001$

- (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`)



$K = 0.001$



$K = 0.01$



$K = 0.1$