

# Image Processing 1 - Exercise 5 - WiSe 2012/13

Weipeng He  
2he@informatik.uni-hamburg.de  
6411529

November 18, 2012

## 1

The original standard deviation is  $\sigma = 4$ , and we expect that the new standard deviation after convolution  $\sigma'$  holds  $(\frac{\sigma'}{2})^2 \leq 12.5\%$ , that is  $\sigma'^2 \leq \frac{1}{2}$ . By using averaging of  $K$  neighbor pixels, we can reduce the standard deviation to

$$\sigma'^2 = \frac{\sigma^2}{K}$$

Thus, we need  $K$  to be

$$K \geq 32$$

Therefore, we can use 7-by-7 mask, of which all values in the matrix is  $\frac{1}{49}$  and  $K = 49 > 32$ .

## 2

As the  $180^\circ$  rotation of the image is still identical to itself, the convolution of the image with itself is the same as the correlation of the image with itself. Thus the grayvalues of the result image would be like a pyramid, namely the center of the square is the brightest, and it becomes darker from the center to margin.

## 3

$$\sin(cx + dy) = \frac{e^{icx}e^{idy} - e^{-icx}e^{-idy}}{2i}$$

Thus

$$\begin{aligned} G(u, v) &= \iint A \sin(cx + dy) e^{-2\pi i(ux + vy)} dx dy \\ &= -\frac{iA}{2} \iint e^{-2\pi ix(u - \frac{c}{2\pi})} e^{-2\pi iy(v - \frac{d}{2\pi})} - e^{-2\pi ix(u + \frac{c}{2\pi})} e^{-2\pi iy(v + \frac{d}{2\pi})} dx dy \\ &= -\frac{iA}{2} \left( \int e^{-2\pi ix(u - \frac{c}{2\pi})} dx \int e^{-2\pi iy(v - \frac{d}{2\pi})} dy - \int e^{-2\pi ix(u + \frac{c}{2\pi})} dx \int e^{-2\pi iy(v + \frac{d}{2\pi})} dy \right) \\ &= -\frac{iA}{2} \left[ \delta(u - \frac{c}{2\pi}, v - \frac{d}{2\pi}) - \delta(u + \frac{c}{2\pi}, v + \frac{d}{2\pi}) \right] \end{aligned}$$

$$\begin{aligned}
G_{uv} &= \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g_{mn} e^{-i2\pi(\frac{mu}{M} + \frac{nv}{N})} \\
&= \frac{1}{M} \sum_{m=0}^{M-1} e^{-i2\pi\frac{mu}{M}} \left( \frac{1}{N} \sum_{n=0}^{N-1} g_{mn} e^{-i2\pi\frac{nv}{N}} \right)
\end{aligned}$$

Therefore, 2-dimensional discrete fourier transform can be decomposited to 2 steps : first transform all rows; then transform all coloumns using the values of last step. In each step 1D-FFT can be applied.

The source code can be found in `src/myfft2.py`.