Exercises for Image Processing 1 - WiSe 2012/13

Exercise 5

1.Chebychev´s Inequality gives a bound for the probability that a random variable r with mean value 0 and standard deviation σ is more than ε away from 0 (It holds for completely arbitrary distributions):

$$P(|r| \geq \varepsilon) \leq (\sigma/\varepsilon)^2$$

Use this inequality for designing a convolution mask which is able to reduce image noise of standard deviation 4 so much, that the probability of noise deviations more than 2 is does not exceed 12.5%.
(2 Points)

2. Image B1 shows an axis-aligned bright square in front of a dark background.

Describe the result of a convolution of the image with itself qualitatively.
(2 Points)

3. Show that the fourier transform of the sinoidal image function $g(x, y) = A \sin(cx + dy)$ is equal to $G(u, v) = -iA/2[\delta(u - c/2\pi, v - d/2\pi) - \delta(u + c/2\pi, v + d/2\pi)]$.
(3 Points)

4. You have a program for computing the 1D-FFT for 2^K values of a real-valued discrete function. How can You use this for doing a 2D-Fourier-Transform of an image of size 512x512 pixels? How can the inverse transform be realized?
Implement your algorithm by using numpy.fft and compare the result with the existing 2D-Fouriertransform. For simplicity use only 2D-Arrays (grayscale images).

Useful numpy.fft-methods (see: http://docs.scipy.org/doc/numpy/reference/routines.fft.html):

- 2D-Fourier-Transform: fi = fft2(i)
- 1D-Fourier-Transform of row k: fz = fft(i[k:k+1, :])
- 1D-Fourier-Transformation of column k: fs = fft(i[:, k:k+1])
- Inverse Transform: ifft resp. ifft2
- (All return a complex valued numpy-Array with real and imaginary part)

(3 Points)