1. m_files

We need to write a couple of m_files.

GenerateGaussian.m

function h = GenerateGaussian (size, lambda)
% this function generates a Gaussian matrix

% generate the meshgrid and the Gaussian matrix
[x, y] = meshgrid([-size/2 : 1 : size/2]);
h = exp(- (x.*x + y.*y) / lambda);

% find the sum and normalize the matrix
sum = sum(sum(h));
h = h ./ sum;

return;

ConvolveImages.m

function out = ConvolveImages (image, h)
% this function convolves two images and
crop the result to the size of the first one

% get the dimensions of the input images
[y_i, x_i] = size(image);
[y_h, x_h] = size(h);

% convolve the images
out = uint8(conv2(image, h));

% crop it to the correct size
out = imcrop(out, [(x_h + 1) / 2, (y_h + 1) / 2, x_i - 1, y_i - 1]);

return;
PSNR.m

function psnr = PSNR(i1, i2)

% calculate PSNR of two images

d = double(i1) - double(i2);
d2 = d .* d;
psnr = 10 * log10(255 * 255 / mean(mean(d2)));

return;

2. Results

Under Matlab, type

    image = imread('image.tif');
    imagesc(image); colormap(gray);

to load and display the image.
h5 = GenerateGaussian(40, 5);
i5 = ConvolveImages(image, h5);
imagesc(i5);
PSNR5 = PSNR(image, i5);

to generate Gaussian matrix with $\lambda = 5$, convolve image with it, display the result and calculate the PSNR against the original.

PSNR5 = 25.1442
Type

\begin{verbatim}
    h10 = GenerateGaussian(40, 10);
    i10 = ConvolveImages(image, h10);
    imagesc(i10);
    PSNR10 = PSNR(image, i10);
\end{verbatim}

to generate Gaussian matrix with \( \lambda = 10 \), convolve image with it, display the result and calculate the PSNR against the original.

PSNR10 = 22.8893