

```

% DSP exercise 15: deconvolution, Oli Lane <ojgl2>, 2014-10-30, MATLAB
clear all; close all;

blurred_orig = im2double(imread('stars-blurred.png'));
psf = im2double(imread('stars-psf.png'));

[originalwidth, originalheight] = size(blurred_orig);

% Extend the original image's edges
padding_amount = 200;
blurred = padarray(blurred_orig, [padding_amount, padding_amount], ...
    'both', 'replicate');

% Create a window to fade the edges of the image with by blurring a
% rectangle
window_padding = 20;
window = ones([originalwidth - 2 * window_padding, ...
    originalheight - 2 * window_padding]);
window = padarray(window, [padding_amount + window_padding, ...
    padding_amount + window_padding], 'both');
window = conv2(window, imresize(psf, 3), 'same');
window = window ./ max(window(:));

% Window the original image to fade the edges out
blurred = blurred .* window;

[m,n] = size(blurred);
[o,p] = size(psf);

% output size
mm = m + o - 1;
nn = n + p - 1;

% Pad the point spread function to the size of the blurred image
psf_padded = padarray(psf, ([mm nn] - [o p]), 'post' );

% Shift the PSF so its centre is 0,0
psf_padded_shifted = circshift(psf_padded, round(-o/2), 1);
psf_padded_shifted = circshift(psf_padded_shifted, round(-p/2), 2);

% Fourier transforms of psf and blurred image
blurred_f = fft2(blurred, mm, nn);
psf_f = fft2(fftshift(psf_padded_shifted), mm, nn);

% Reduce the noise a bit
epsilon = 1;
for i = 1:numel(psf_f)
    if(abs(psf_f(i)) < epsilon)

```

```
        psf_f(i) = epsilon;
    end
end

% Do the deconvolution and truncate the padding we added earlier
div = blurred_f ./ (psf_f);
reconstructed = ifftshift(ifft2(div));
reconstructed = reconstructed(padding_amount:originalwidth+padding_amount, ...
                             padding_amount:originalheight+padding_amount);

reconstructed = reconstructed * 150;

% Show the original
figure(1); imshow(blurred_orig)
% Show the result
figure(2); imshow(reconstructed)
```

