
Part 2: Notch Filter

Image: moonlanding

```
image = (imread("./images/moonlanding.png"));
image = im2double(image);

% Apply notch filter
filteredImg = notchFilter(image, [0.5, 1]);

figure;
imshow(image);
title('Origin');

figure;
imshow(filteredImg, []);
title('Noise removed');

% Image: psnr2

image = rgb2gray(imread("./images/psnr2.png"));
image = im2double(image);

% Apply notch filter
filteredImg = notchFilter(image,[0.75, 1]);

figure;
imshow(image);
title('Origin');

figure;
imshow(filteredImg, []);
title('Noise removed');

function res = notchFilter(image, range)
    % Calcualte DFT
    fftImg = fftshift(fft2(image));
    amp = abs(fftImg);

    minAmp = min(amp(:));
    s = sort(amp(:));
    maxAmp = s(end - 20);

    % Calculate lower bound and upper bound
    lowerbound = (maxAmp - minAmp) * range(1) + minAmp;
    upperbound = (max(amp(:)) - minAmp) * range(2) + minAmp;
    band = (amp >= lowerbound) & (amp < upperbound);

    peak = (amp == ordfilt2(amp, 9 , ones(3, 3))) & band;
    [w, h] = size(image);
    [r, c] = find(peak);
```

```

removeLength = 5;

noise = zeros(size(fftImg));
for i = 1:length(r)
    % Edge case
    if (w-r(i))^2+(h-c(i))^2 <= removeLength^2
        continue;
    end
    if r(i) <= removeLength || c(i) <= removeLength
        continue;
    end
    % Calculate the end of rows and cols
    if w < r(i) + removeLength
        rowEnd = w;
    else
        rowEnd = r(i) + removeLength;
    end
    if h < c(i) + removeLength
        colEnd = h;
    else
        colEnd = c(i) + removeLength;
    end

    dr = r(i) - removeLength:rowEnd;
    dc = c(i) - removeLength:colEnd;

    noise(dr, dc) = 1;
end

% Get the central of the image
cx = round((size(noise, 2) + 1) / 2);
cy = round((size(noise, 1) + 1) / 2);

dcr = cy - 2*removeLength:cy + 2*removeLength;
drr = cx - removeLength:cx + 2*removeLength;
noise(dcr, drr) = 0;

fftImg(noise > 0) = 0;

showResult(amp, fftImg);

res = real(ifft2(fftshift(fftImg)));
end

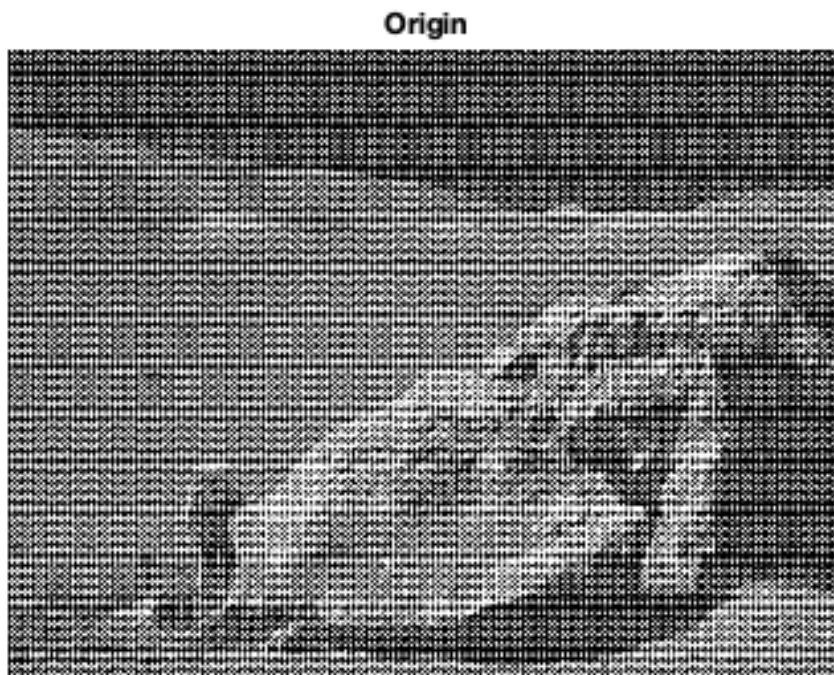
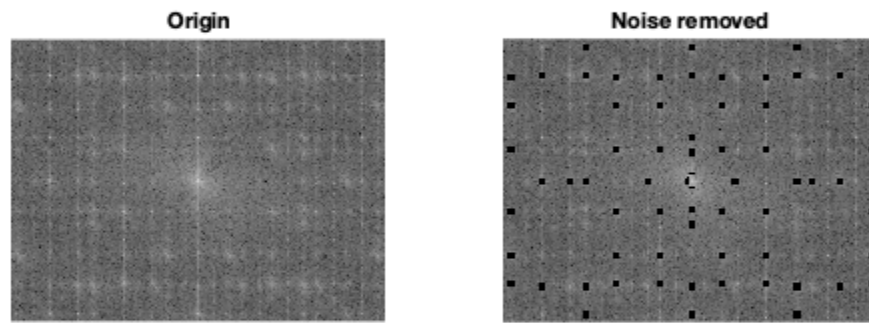
function result = showResult(amp, fftImg)
    figure;

    subplot(121)
    imshow(log(amp),[]);
    title("Origin");

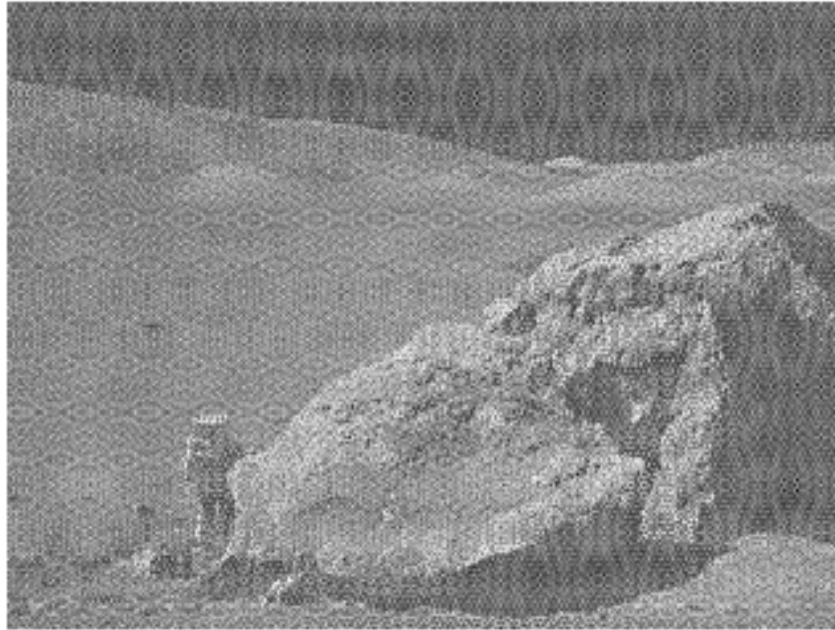
    subplot(122)
    imshow(real(log(abs(fftImg))), []);
    title('Noise removed');

```

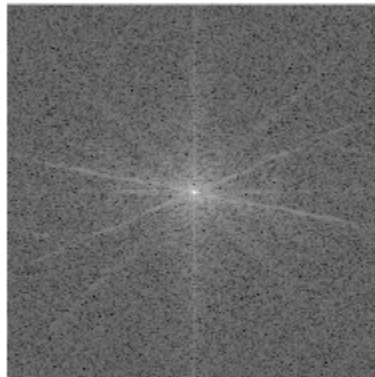
end



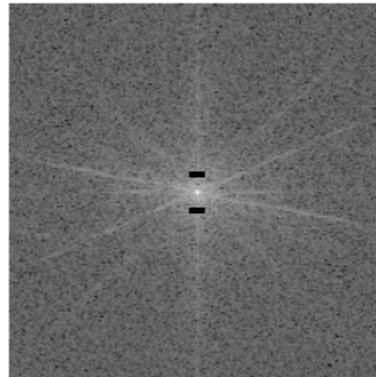
Noise removed



Origin



Noise removed



Origin



Noise removed



Published with MATLAB® R2019b