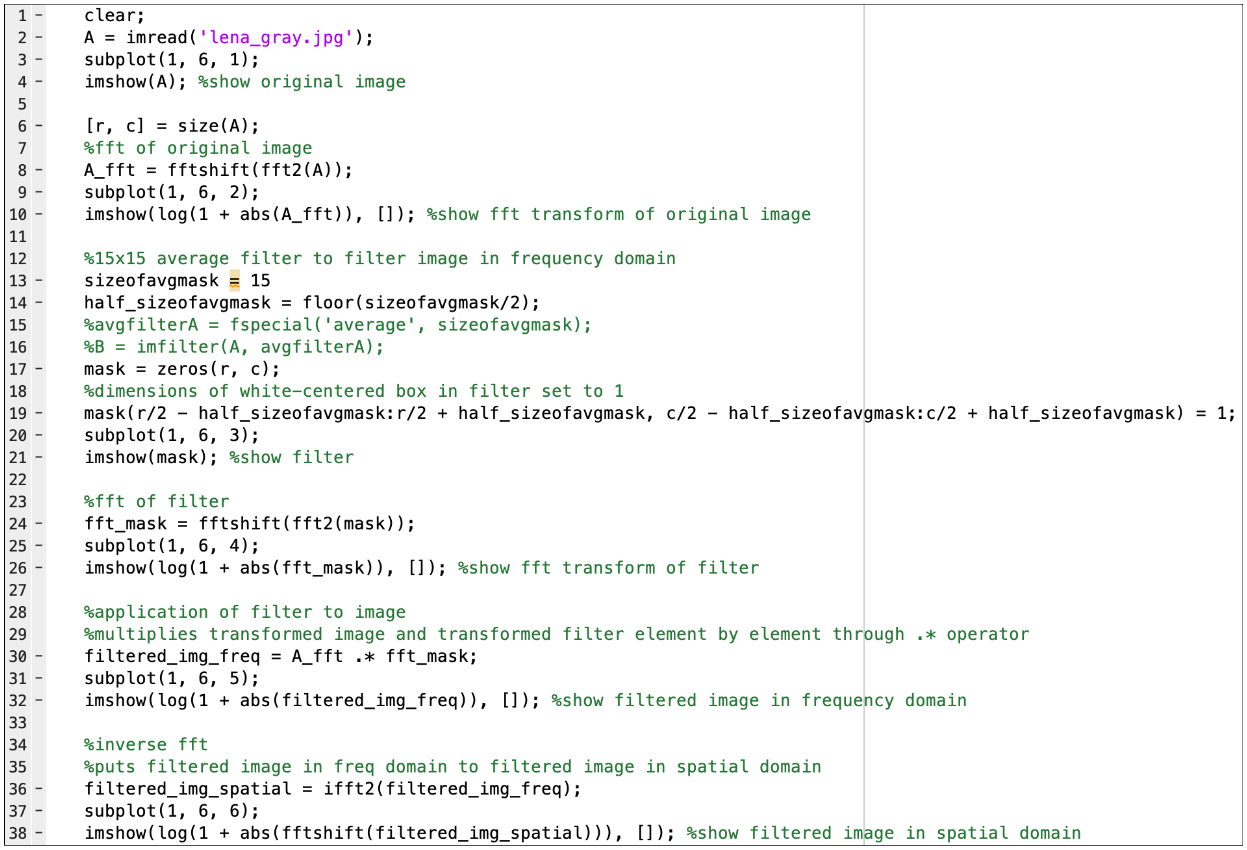
Haruka Kido
EE456: Digital Image Processing
Assignment 3

1. Filters applied to “lena_gray.jpg” image in the **frequency domain**:

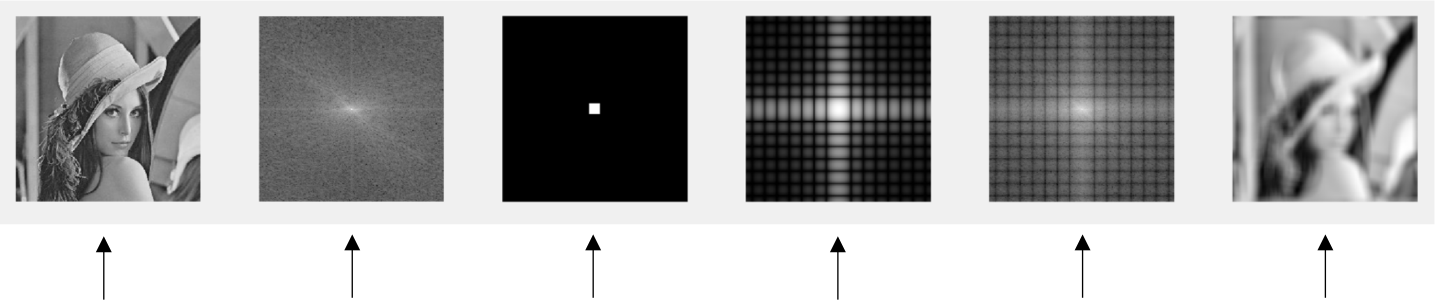
   a. 15x15 Average filter:

      ○ Source code:



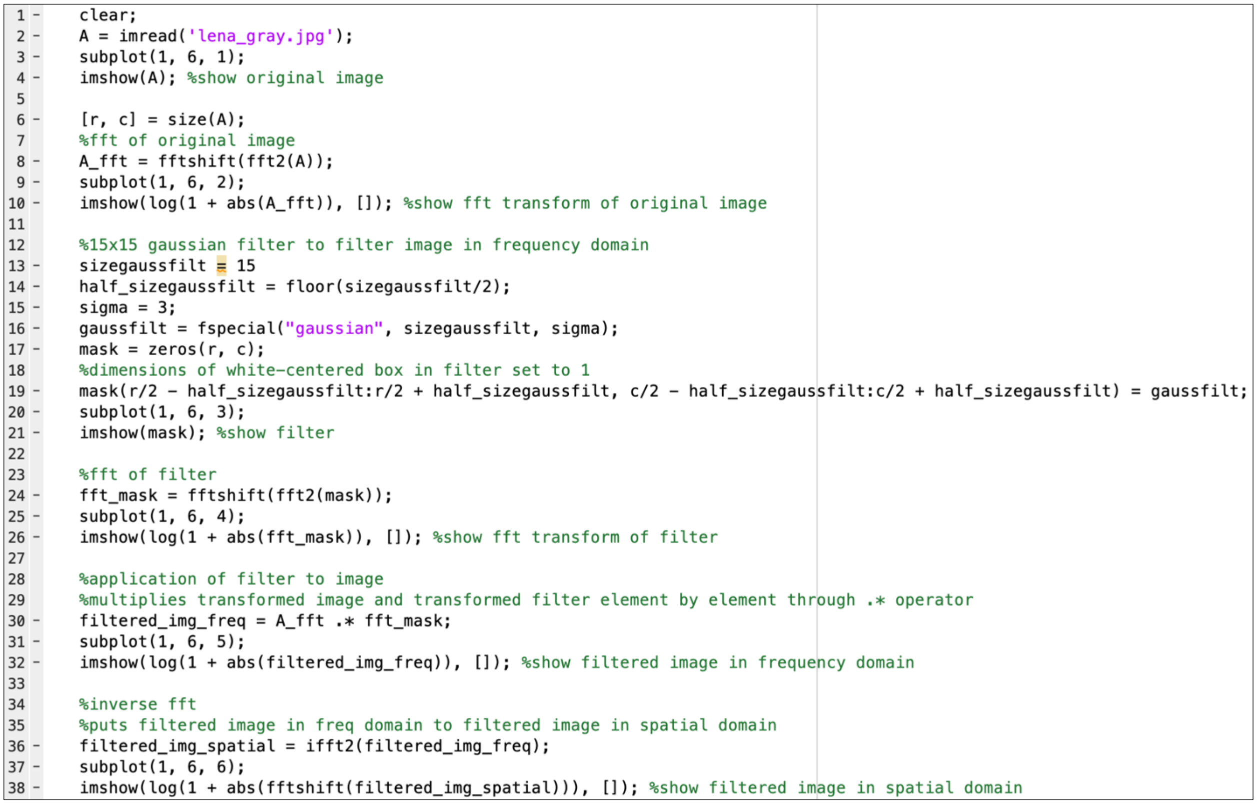
```
clear;
A = imread('lena_gray.jpg');
subplot(1, 6, 1);
imshow(A); %show original image

[r, c] = size(A);
%fft of original image
A_fft = fftshift(fft2(A));
subplot(1, 6, 2);
imshow(log(1 + abs(A_fft)), []); %show fft transform of original image

%15x15 average filter to filter image in frequency domain
sizeofavgmask = 15
half_sizeofavgmask = floor(sizeofavgmask/2);
%avgfilterA = fspecial('average', sizeofavgmask);
%B = imfilter(A, avgfilterA);
mask = zeros(r, c);
%dimensions of white-centered box in filter set to 1
mask(r/2 - half_sizeofavgmask:r/2 + half_sizeofavgmask, c/2 - half_sizeofavgmask:c/2 + half_sizeofavgmask) = 1;
subplot(1, 6, 3);
imshow(mask); %show filter

%fft of filter
fft_mask = fftshift(fft2(mask));
subplot(1, 6, 4);
imshow(log(1 + abs(fft_mask)), []); %show fft transform of filter

%application of filter to image
%multiplies transformed image and transformed filter element by element through .* operator
filtered_img_freq = A_fft .* fft_mask;
subplot(1, 6, 5);
imshow(log(1 + abs(filtered_img_freq)), []); %show filtered image in frequency domain

%inverse fft
%puts filtered image in freq domain to filtered image in spatial domain
filtered_img_spatial = ifft2(filtered_img_freq);
subplot(1, 6, 6);
imshow(log(1 + abs(fftshift(filtered_img_spatial))), []); %show filtered image in spatial domain
```

      ○ Original image and its FFT transform
      ○ Filter and its FFT transform
      ○ Filtered image in frequency domain
      ○ Filtered image in spatial domain


Original image
FFT transform of original image
15x15 averaging filter
FFT transform of filter
Filtered image (in frequency domain)
Filtered image (in spatial domain)

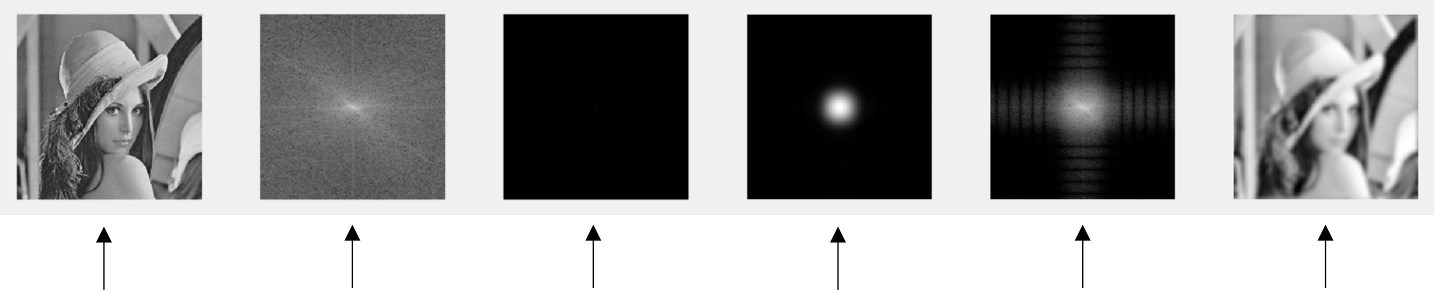
b. 15x15 Gaussian filter:

- Source code:



```
clear;
A = imread('lena_gray.jpg');
subplot(1, 6, 1);
imshow(A); %show original image

[r, c] = size(A);
%fft of original image
A_fft = fftshift(fft2(A));
subplot(1, 6, 2);
imshow(log(1 + abs(A_fft)), []); %show fft transform of original image

%15x15 gaussian filter to filter image in frequency domain
sizegaussfilt = 15
half_sizegaussfilt = floor(sizegaussfilt/2);
sigma = 3;
gaussfilt = fspecial("gaussian", sizegaussfilt, sigma);
mask = zeros(r, c);
%dimensions of white-centered box in filter set to 1
mask(r/2 - half_sizegaussfilt:r/2 + half_sizegaussfilt, c/2 - half_sizegaussfilt:c/2 + half_sizegaussfilt) = gaussfilt;
subplot(1, 6, 3);
imshow(mask); %show filter

%fft of filter
fft_mask = fftshift(fft2(mask));
subplot(1, 6, 4);
imshow(log(1 + abs(fft_mask)), []); %show fft transform of filter

%application of filter to image
%multiplies transformed image and transformed filter element by element through .* operator
filtered_img_freq = A_fft .* fft_mask;
subplot(1, 6, 5);
imshow(log(1 + abs(filtered_img_freq)), []); %show filtered image in frequency domain

%inverse fft
%puts filtered image in freq domain to filtered image in spatial domain
filtered_img_spatial = ifft2(filtered_img_freq);
subplot(1, 6, 6);
imshow(log(1 + abs(fftshift(filtered_img_spatial))), []); %show filtered image in spatial domain
```

- Original image and its FFT transform
- Filter and its FFT transform
- Filtered image in frequency domain
- Filtered image in spatial domain



Original image

FFT transform of
original image

15x15 Gaussian filter

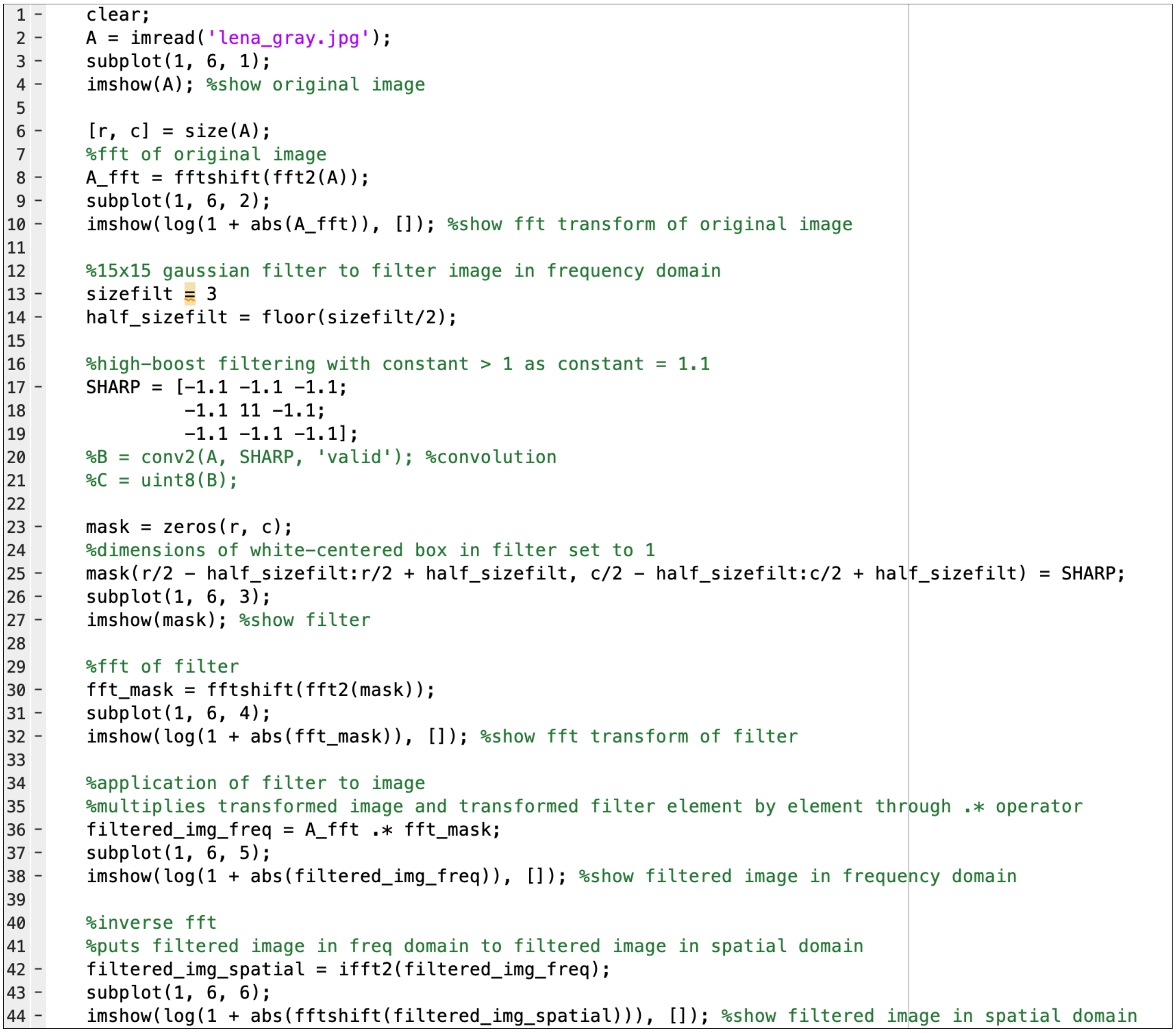
FFT transform of filter

Filtered image
(in frequency domain)

Filtered image
(in spatial domain)

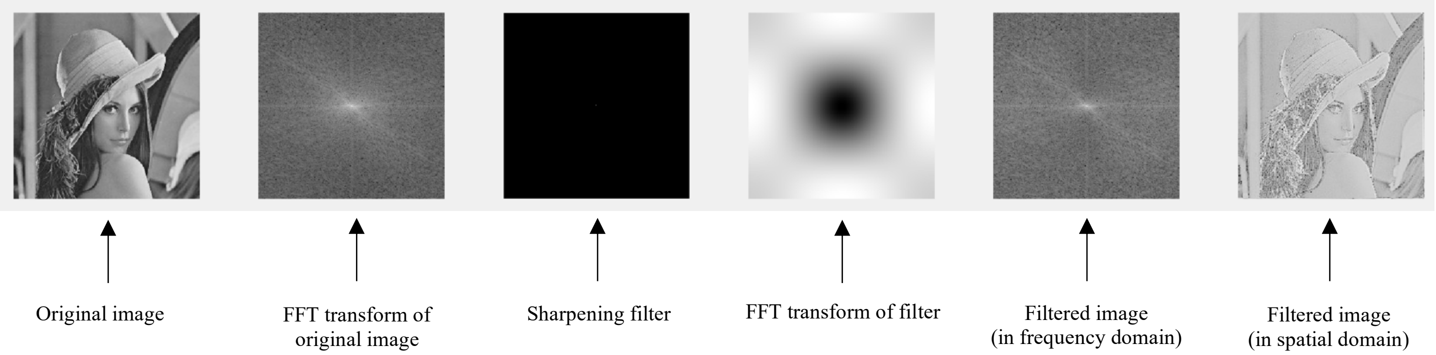
c. Any Sharpening filter:

- Source code:



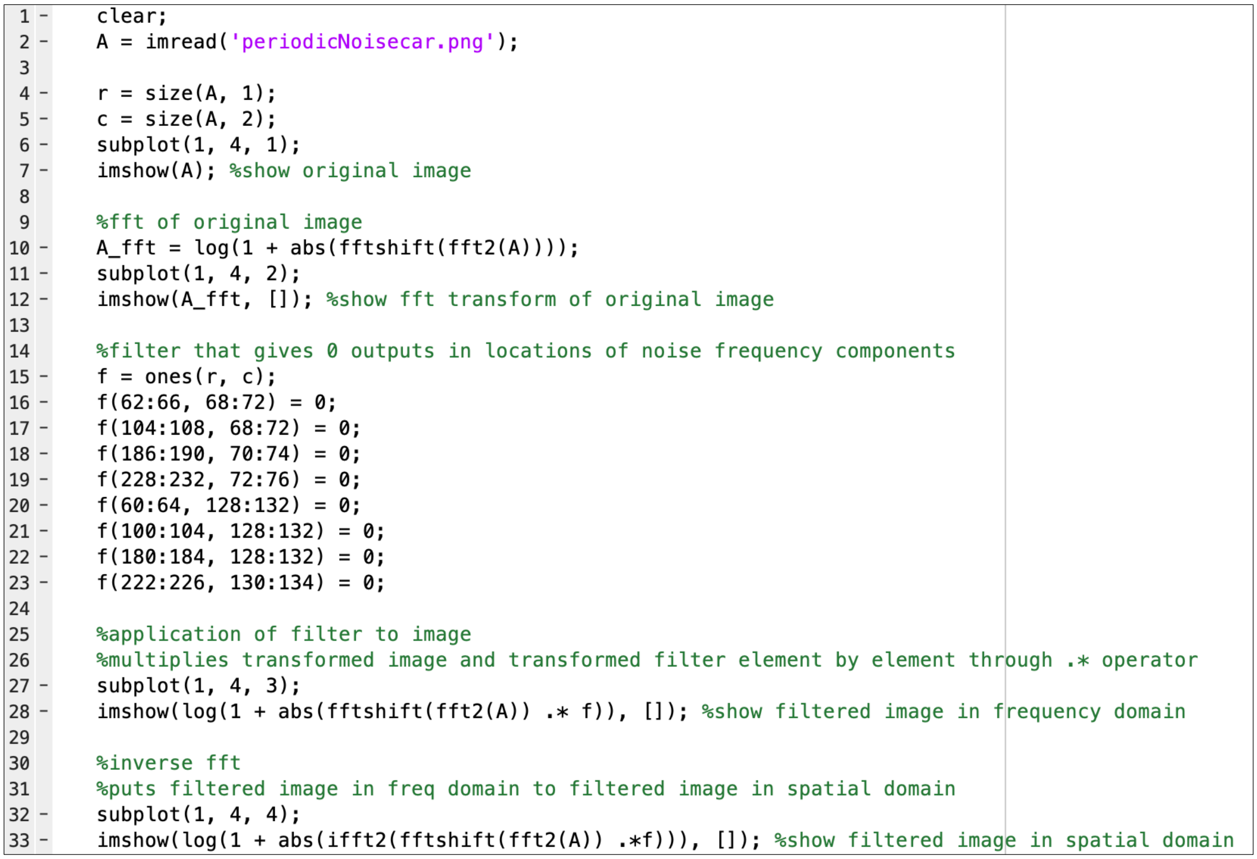
```
clear;
A = imread('lena_gray.jpg');
subplot(1, 6, 1);
imshow(A); %show original image

[r, c] = size(A);
%fft of original image
A_fft = fftshift(fft2(A));
subplot(1, 6, 2);
imshow(log(1 + abs(A_fft)), []); %show fft transform of original image

%15x15 gaussian filter to filter image in frequency domain
sizefilt = 3
half_sizefilt = floor(sizefilt/2);

%high-boost filtering with constant > 1 as constant = 1.1
SHARP = [-1.1 -1.1 -1.1;
         -1.1 11 -1.1;
         -1.1 -1.1 -1.1];
%B = conv2(A, SHARP, 'valid'); %convolution
%C = uint8(B);

mask = zeros(r, c);
%dimensions of white-centered box in filter set to 1
mask(r/2 - half_sizefilt:r/2 + half_sizefilt, c/2 - half_sizefilt:c/2 + half_sizefilt) = SHARP;
subplot(1, 6, 3);
imshow(mask); %show filter

%fft of filter
fft_mask = fftshift(fft2(mask));
subplot(1, 6, 4);
imshow(log(1 + abs(fft_mask)), []); %show fft transform of filter

%application of filter to image
%multiplies transformed image and transformed filter element by element through .* operator
filtered_img_freq = A_fft .* fft_mask;
subplot(1, 6, 5);
imshow(log(1 + abs(filtered_img_freq)), []); %show filtered image in frequency domain

%inverse fft
%puts filtered image in freq domain to filtered image in spatial domain
filtered_img_spatial = ifft2(filtered_img_freq);
subplot(1, 6, 6);
imshow(log(1 + abs(fftshift(filtered_img_spatial))), []); %show filtered image in spatial domain
```

- Original image and its FFT transform
- Filter and its FFT transform
- Filtered image in frequency domain
- Filtered image in spatial domain


Original image
FFT transform of original image
Sharpening filter
FFT transform of filter
Filtered image (in frequency domain)
Filtered image (in spatial domain)

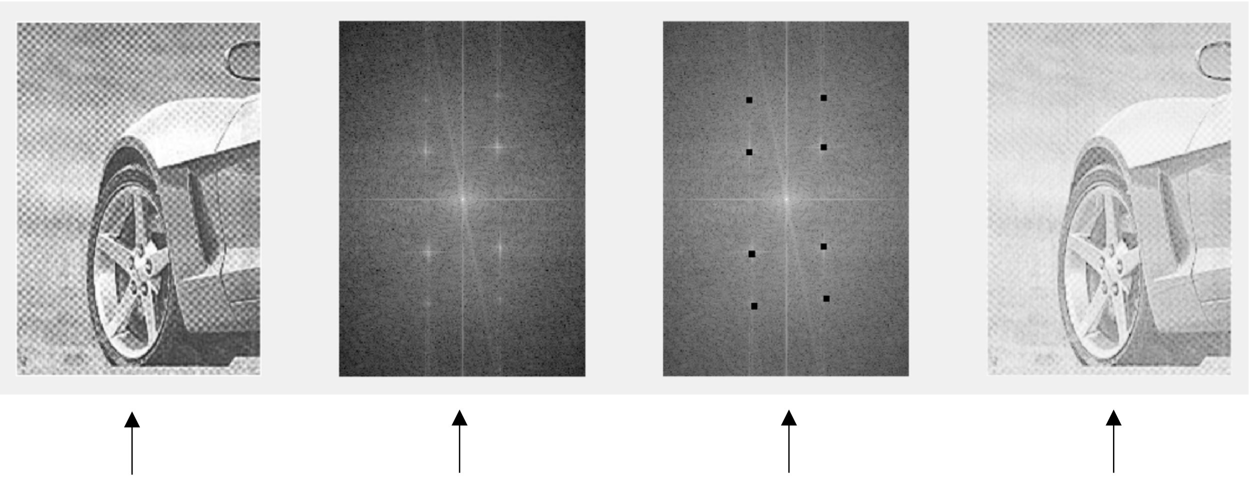
2. Procedure in MATLAB to reduce the periodic noise on the background of "periodicNoiseCar.png" image:

   ○ Source code:



```
clear;
A = imread('periodicNoisecar.png');

r = size(A, 1);
c = size(A, 2);
subplot(1, 4, 1);
imshow(A); %show original image

%fft of original image
A_fft = log(1 + abs(fftshift(fft2(A))));
subplot(1, 4, 2);
imshow(A_fft, []); %show fft transform of original image

%filter that gives 0 outputs in locations of noise frequency components
f = ones(r, c);
f(62:66, 68:72) = 0;
f(104:108, 68:72) = 0;
f(186:190, 70:74) = 0;
f(228:232, 72:76) = 0;
f(60:64, 128:132) = 0;
f(100:104, 128:132) = 0;
f(180:184, 128:132) = 0;
f(222:226, 130:134) = 0;

%application of filter to image
%multiplies transformed image and transformed filter element by element through .* operator
subplot(1, 4, 3);
imshow(log(1 + abs(fftshift(fft2(A)) .* f)), []); %show filtered image in frequency domain

%inverse fft
%puts filtered image in freq domain to filtered image in spatial domain
subplot(1, 4, 4);
imshow(log(1 + abs(ifft2(fftshift(fft2(A)) .*f))), []); %show filtered image in spatial domain
```

   ○ Original image and its FFT transform
   ○ Filtered image in frequency domain
   ○ Filtered image in spatial domain



Original image

FFT transform of original image

Filtered image (in frequency domain)

Filtered image (in spatial domain)