Image processing for bioinformatics

Laboratory
Spatial filtering and filtering in Fourier space

1 Examples

1.1 Spatial filtering

Code

```
%% Average, gaussian and median filter **********
Img = imread('eighttif.png'); Img = im2double(Img(:,:,1));
% Add noise
Img_spn = imnoise(Img,'salt & pepper',0.1); % noise density
Img_gn = imnoise(Img,'gaussian',0,0.01); % M,V
% Kernels and filtering
h_b = fspecial('average',5); % hsize
Img_spn_boxF = imfilter(Img_spn,h_b,'symmetric','corr');
Img_gn_boxF = imfilter(Img_gn,h_b,'replicate','corr');

h_g = fspecial('gaussian',5,1); % hsize,sigma
Img_spn_gaussianF = imfilter(Img_spn,h_g,'replicate','corr');
Img_gn_gaussianF = imfilter(Img_gn,h_g,'circular','corr');

Img_spn_medF = medfilt2(Img_spn,[3 3],'symmetric' ); % Img, [Neighborhood size], 'Padding'
Img_gn_medF = medfilt2(Img_gn,[3 3],'symmetric' );
```

Image

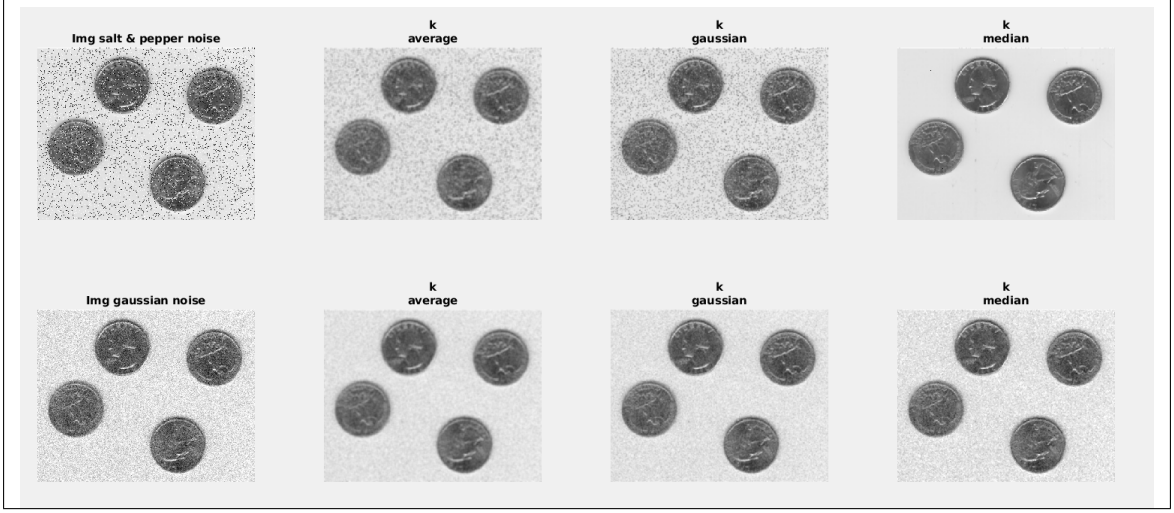

Table 1: Spatial filtering

1.2 Fourier Transform

Code

```
% Fourier Transform
Img = imread('eighttif.png');
Img = im2double(Img(:,:,1));
Img_ff = fft2(Img);
Img_ff_s = fftshift(Img_ff);
Img_iff = real(ifft2(Img_ff));
```

Image

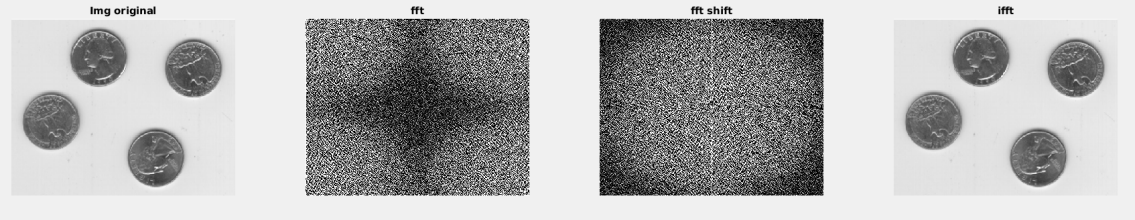

Table 2: Fourier Transform

2 Assignment

1. Implement the general filter equation (correlation)

$$I'(u,v) = \sum_{(i,j)\in R_H} I(u+i, v+j) \cdot H(i,j)$$

   and reproduce the examples of Table 1.

2. Implement filtering in Fourier space (convolution).

Image

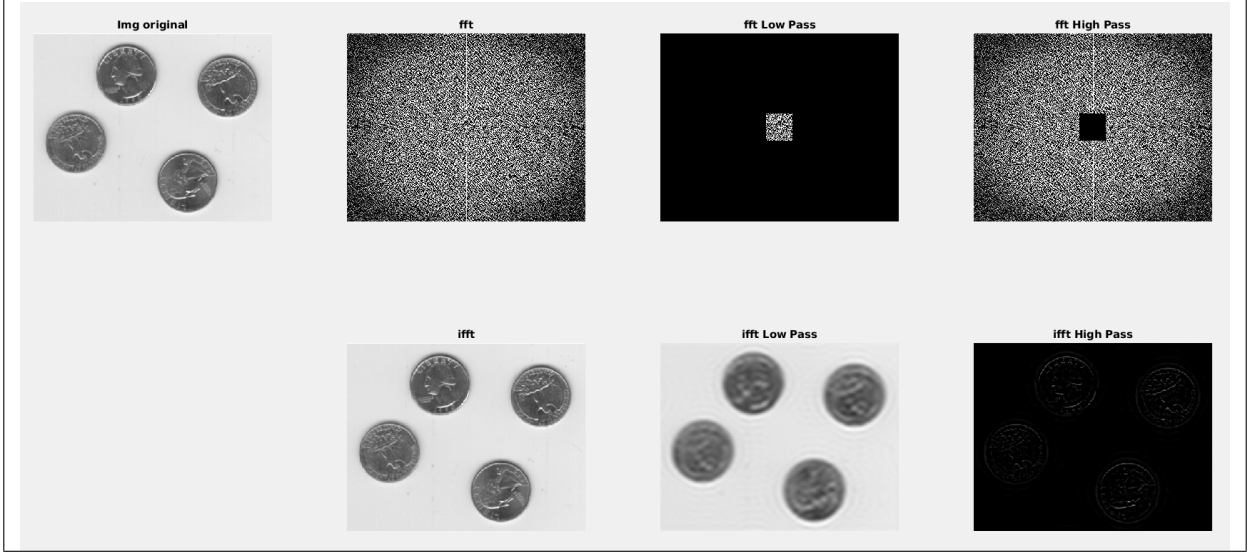

Table 3: Filtering in Fourier space

3 Solutions

1. Implement the general filter equation (correlation)

$$I'(u,v) = \sum_{(i,j)\in R_H} I(u+i, v+j) \cdot H(i,j)$$

Function: Correlation_kernel

```
%% Implement the general filter equation (correlation) **********
function mat = Correlation_kernel(Img, k)
  [m,n] = size(k);
  Rm = (m-1)/2;
  kCenterM = ceil(Rm+1);

  Rn = (n-1)/2;
  kCenterN = ceil(Rn+1);

  %Pad Img
  [m,n] = size(Img);
  ImgZ = zeros(m+2*Rm,n+2*Rn); ImgZ(1+Rm:m+Rm,1+Rn:n+Rn) = Img;

  [m,n] = size(ImgZ);
  matCorrelation = zeros(m,n);

  for u = 1+Rm:m-Rm
    for v = 1+Rn:n-Rn
      tmp = 0;
      tmp2 = 0;
      for i = -Rm:Rm
        for j = -Rn:Rn
          tmp = tmp + ImgZ(u+i,v+j) * k(kCenterM + i,kCenterN + j);
          tmp2 = tmp2 + ImgZ(u-i,v-j) * k(kCenterM + i,kCenterN + j);
        end
      end
      matCorrelation(u,v) = tmp;
    end
  end
  [m,n] = size(Img);
  mat = matCorrelation(1+Rm:m+Rm,1+Rn:n+Rn);
end
```

2. Implement filtering in Fourier space (convolution).

Code - Filtering in Fourier space

```
%% Implement filtering in Fourier space ********************************
clc; clear; close all;
Img = imread('eighttif.png');
Img = im2double(Img(:,:,1));
[m,n] = size(Img);

Img_ff = fftshift(fft2(Img));

hf = 25;
mask_LowPass = zeros(m,n);
mask_LowPass(m/2-hf:m/2+hf,n/2-hf:n/2+hf) = 1;

mask_HighPass = ones(m,n);
mask_HighPass(m/2-hf:m/2+hf,n/2-hf:n/2+hf) = 0;
% filtering
Img_ff_lp = Img_ff.*mask_LowPass;
Img_ff_hp = Img_ff.*mask_HighPass;

Img_iff = ifft2(ifftshift(Img_ff));
Img_iff_lp = ifft2(ifftshift(Img_ff_lp));
Img_iff_hp = ifft2(ifftshift(Img_ff_hp));

nr = 2;
nc = 4;
subplot(nr,nc,1); imshow(Img,[0,1]); title('Img original');

subplot(nr,nc,2); imshow(Img_ff,[0,1]); title('fft');
subplot(nr,nc,3); imshow(Img_ff_lp,[0,1]); title('fft Low Pass');
subplot(nr,nc,4); imshow(Img_ff_hp,[0,1]); title('fft High Pass');

subplot(nr,nc,6); imshow(Img_iff,[0,1]); title('ifft');
subplot(nr,nc,7); imshow(Img_iff_lp,[0,1]); title('ifft Low Pass');
subplot(nr,nc,8); imshow(Img_iff_hp,[0,1]); title('ifft High Pass');
```