School of Data and Computer Science, Sun Yat-sen University

PROJECT 04-01 $\sim$ 04-05 AND Rotation Research

17341137 Zhenpeng Song

October 8, 2019

Abstract

Implementation of 5 experiments:

- I - Two-Dimensional Fast Fourier Transform.
- II - Fourier Spectrum and Average Value.
- III - Lowpass Filtering.
- IV - Highpass Filtering Using a Lowpass Image.
- V - Correlation in the Frequency Domain.
- VI - Do a research on the characteristic of the spectrum, which means the spectrum will rotate the same degree as the image does.

1 Technical discussion

- I - Two-Dimensional Fast Fourier Transform.
  There are 5 missions in this part:

  1. (a) To centralize the image, use meshgrid() in MATLAB to generate a grid to do so for each pixel in the image.
  2. (b) The 'real function' turns out to be an value of a real number.
  3. (c) Call ifft2() in MATLAB to do so.
  4. (d) To Centralize the real part of the result.
  5. (e) Use abs() to calculate the module of the complex to obtain the spectrum.

- II - Fourier Spectrum and Average Value.
  This PJ. can be explain as following steps:

  1. Centralize the input image.
  2. Do FFT on the result.(With fft2() in MATLAB).
  3. Calculate the spectrum.(With abs() in MATLAB).

  In addition, since the spectrum I had generated turn out to be a bright graph, I do a Gamma-Transformation to get a clearer graph.

- III - Lowpass Filtering.
  Using the equation:
  $$H(u,v) = \mathrm{e}^{-D^2(u,v)/2\sigma^2}$$
  And do as follow:

  1. Calculate the filter $H(u,v)$.
  2. Do FFT on the centralized image.
  3. Multiply the FFT result and $H(u,v)$.
  4. Do IFFT on the result.
  5. Centralize the real part of the result.

- IV - Highpass Filtering Using a Lowpass Image.
  Usign the equation:
  $$F(u,v) - F(u,v)H(u,v) = F(u,v)(1 - H(u,v)) = F(u,v)G(u,v)$$
  We can simply subtract the lowpass filtered image from the origin image to obtain the highpass result.

- V - Correlation in the Frequency Domain.
  Using the equation:
  $$f(x,y) \circ h(x,y) \Longleftrightarrow F^{\star}(u,v)H(u,v)$$
  And do as follow, for two target images I1, I2:

  1. Centralize I1, I2.
  2. Do FFT on each of the result.
  3. Multiply the DFT of I2 by the imaginary part of the DFT of I1.

4. Do IFFT on the result.
5. Centralize the real part of the result.

- VI - Do a research on the characteristic of the spectrum, which means the spectrum will rotate the same degree as the image does.
  Use imrotate() in MATLAB to rotate the image, and generate the spectrums of different degree. Then observe the results.

2 Discussion

- Explanation for PJ04-04
  Since the sigma in lowpass filter will affect and limit the area the filter covers. Consequently, with a small sigma, much highpass components will not be filtered, leaving many detailed features.
- Explanation for PJ04-05
  The (x, y) mentioned in the project is $(191, 193)$.

3 Results

- I - Two-Dimensional Fast Fourier Transform.

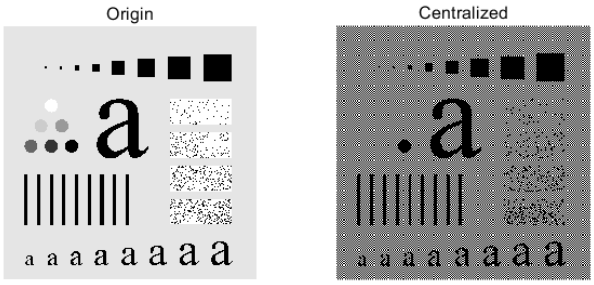

Figure 1: PJ0401

- II - Fourier Spectrum and Average Value.

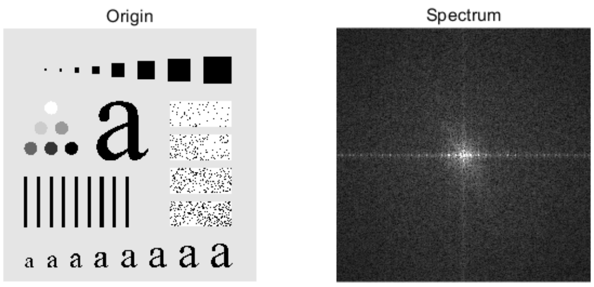

Figure 2: PJ0402

- III - Lowpass Filtering.

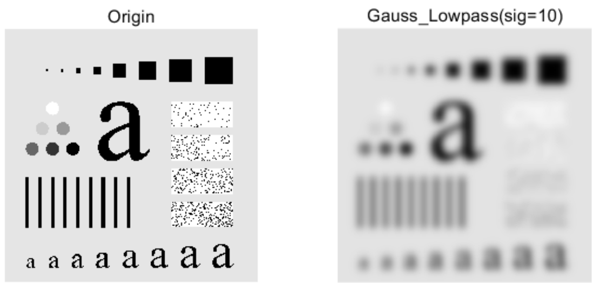

Figure 3: PJ0403

- IV - Highpass Filtering Using a Lowpass Image.

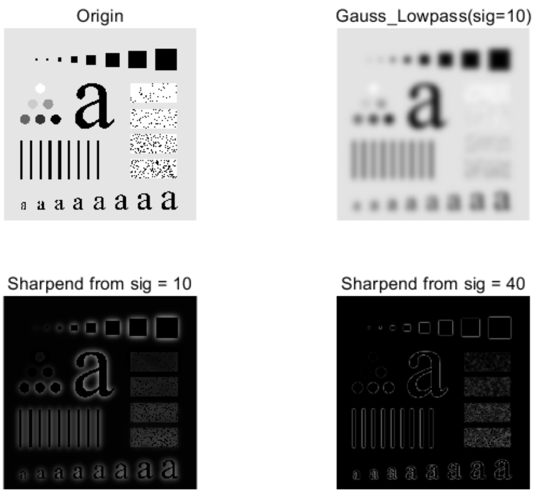

Figure 4: PJ0404

- V - Correlation in the Frequency Domain.

Figure 5: PJ0405

- VI - Do a research on the characteristic of the spectrum, which means the spectrum will rotate the same degree as the image does.

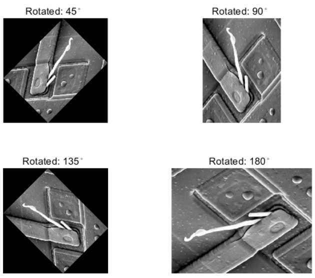

Figure 6: Rotation - 1

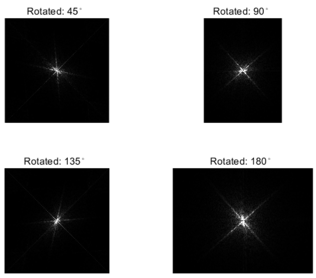

Figure 7: Rotation - 2

4 Appendix

```
% PJ0401
close all; clear, clc;
img1 = imread('./Fig4.04(a).jpg');
img2 = imread('./Fig4.41(a).jpg');
img3 = imread('./Fig4.41(b).jpg');
img4 = imread('./Fig0418(a).tif');

% a
figure(1);
subplot(121);
imshow(img4); title('Origin');
subplot(122);
imshow(centralize(img4)); title('Centralized');

% b
% Refer to multi_complex.m

% c
% Refer to IFFT.m

% d
% Refer to centralizeR.m

% e
% Refer to spectrum.m

% PJ0402
close all; clear, clc;
img1 = imread('./Fig4.04(a).jpg');
img2 = imread('./Fig4.41(a).jpg');
img3 = imread('./Fig4.41(b).jpg');
img4 = imread('./Fig0418(a).tif');

% a
spec = spectrum(fft2(centralize(double(img4))));

% b
figure(1);
subplot(121); imshow(img4); title('Origin');
subplot(122); imshow(uint8(spec.^0.4)); title('Spectrum');

% c
avg = mean(img4, 'all');
disp("average value of img4:");
disp(avg);

```

```
% PJ0403
close all; clear, clc;
img1 = imread('./Fig4.04(a).jpg');
img2 = imread('./Fig4.41(a).jpg');
img3 = imread('./Fig4.41(b).jpg');
img4 = imread('./Fig0418(a).tif');

% a
% Refer to gauss_lowpass.m

% b
[M, N] = size(img4);
gauss_lp = gauss_lowpass(img4, M/2, N/2, 10);
imwrite(uint8(gauss_lp), './Fig0418(b).tif');

figure(1);
subplot(121);
imshow(img4); title('Origin');
subplot(122);
imshow(uint8(gauss_lp)); title('Gauss\_Lowpass(sig=10)');


% PJ0404
close all; clear, clc;
img1 = imread('./Fig4.04(a).jpg');
img2 = imread('./Fig4.41(a).jpg');
img3 = imread('./Fig4.41(b).jpg');
img4 = imread('./Fig0418(a).tif');
lp_img4 = imread('./Fig0418(b).tif');

% a
s_img1 = img4 - lp_img4;

% b
[M, N] = size(img4);
gauss_lp = gauss_lowpass(img4, M/2, N/2, 40);
s_img2 = img4 - uint8(gauss_lp);

figure(1);
subplot(221);
imshow(img4); title('Origin');
subplot(222);
imshow(lp_img4); title('Gauss\_Lowpass(sig=10)');
subplot(223);
imshow(uint8(s_img1)); title('Sharpend from sig = 10');
subplot(224);
imshow(uint8(s_img2)); title('Sharpend from sig = 40');

```

```matlab

% PJ0405
close all; clear, clc;
img1 = imread('./Fig4.04(a).jpg');
img2 = imread('./Fig4.41(a).jpg');
img3 = imread('./Fig4.41(b).jpg');
img4 = imread('./Fig0418(a).tif');

[M1, N1] = size(img2);
[M2, N2] = size(img3);

I1 = zeros(298, 298);
I2 = zeros(298, 298);
I1(1:M1, 1:N1) = img2(1:M1, 1:N1);
I2(1:M2, 1:N2) = img3(1:M2, 1:N2);

cor = fft2(centralize(I2)) .* conj(fft2(centralize(I1)));
res = centralizeR(IFFT(cor));

figure(1);
subplot(131);
imshow(img2); title('Origin');
subplot(132);
imshow(img3); title('Mask');
subplot(133);
imshow(mat2gray(res), []); title('Correlation');

max_val = max(max(res));
[r, c] = find(res == max_val);
disp("Max value:"); disp(max_val);
disp(" at "); disp([r, c]);


% PJ0406
close all; clear, clc;
img1 = imread('./Fig4.04(a).jpg');

I45 = rotate(img1, 45);
I90 = rotate(img1, 90);
I135 = rotate(img1, 135);
I180 = rotate(img1, 180);

figure(1);
subplot(221); imshow(I45); title('Rotated: 45^ ');
subplot(222); imshow(I90); title('Rotated: 90^ ');
subplot(223); imshow(I135); title('Rotated: 135^ ');
subplot(224); imshow(I180); title('Rotated: 180^ ');

S45 = spectrum(IFFT(centralize(I45)));
```

```
S90 = spectrum(IFFT(centralize(I90)));
S135 = spectrum(IFFT(centralize(I135)));
S180 = spectrum(IFFT(centralize(I180)));

figure(2);
subplot(221); imshow(S45); title('Rotated: 45^ ');
subplot(222); imshow(S90); title('Rotated: 90^ ');
subplot(223); imshow(S135); title('Rotated: 135^ ');
subplot(224); imshow(S180); title('Rotated: 180^ ');


function res = centralize(img)
    [M, N] = size(img);
    [Y, X] = meshgrid(1:N, 1:M);
    c = (-1).^(X + Y);
    res = double(img) .* c;
end

function res = centralizeR(A)
    res = centralize(real(A));
end

function res = spectrum(A)
    res = abs(A);
end

function res = IFFT(A)
    res = ifft2(A);
end

function res = multi_complex(C, r)
    res = C * r;
end

function res = gauss_lowpass(img, fc_x, fc_y, sig)
    [M, N] = size(img);
    [Y, X] = meshgrid(1:N, 1:M);
    D = (X - fc_x).^2 + (Y - fc_y).^2;
    H = exp(-D/(2*sig^2));
    res = centralizeR(IFFT(H .* fft2(centralize(img))));
end

function res = rotate(img, deg)
    res = imrotate(img, deg);
end
```