Colin Keenan

ECE 5470

3-9-2020

Homework 6

1. High Pass Filtering

# Ideal Filter

clear

imageA = im2double(imread("Fig6-1a.jpg"));

imageB = im2double(imread("Fig6-1b.jpg"));

D0\_1 =14;

D0\_2 =70;

A\_IHPF\_1 = ihpf(imageA, D0\_1);

A\_IHPF\_2 = ihpf(imageA, D0\_2);

B\_IHPF\_1 = ihpf(imageB, D0\_1);

B\_IHPF\_2 = ihpf(imageB, D0\_2);

A\_HP\_1 = imageA + A\_IHPF\_1;

A\_HP\_2 = imageA + A\_IHPF\_2;

B\_HP\_1 = imageB + B\_IHPF\_1;

B\_HP\_2 = imageB + B\_IHPF\_2;

figure();

% Image A

subplot(2,5,1)

imshow(imageA);

title("Fig6-1a.jpg");

subplot(2,5,2)

imshow(A\_IHPF\_1, []);

title("IHPF(" + D0\_1 + ")");

subplot(2,5,3)

imshow(A\_HP\_1);

title("After Filter(" + D0\_1 + ")");

subplot(2,5,4)

imshow(A\_IHPF\_2, []);

title("IHPF(" + D0\_2 + ")");

subplot(2,5,5)

imshow(A\_HP\_2);

title("After Filter(" + D0\_2 + ")");

% Image B

subplot(2,5,6)

imshow(imageB);

title("Fig6-1b.jpg");

subplot(2,5,7)

imshow(B\_IHPF\_1, []);

title("IHPF(" + D0\_1 + ")");

subplot(2,5,8)

imshow(B\_HP\_1);

title("After Filter(" + D0\_1 + ")");

subplot(2,5,9)

imshow(B\_IHPF\_2, []);

title("IHPF(" + D0\_2 + ")");

subplot(2,5,10)

imshow(B\_HP\_2);

title("After Filter(" + D0\_2 + ")");

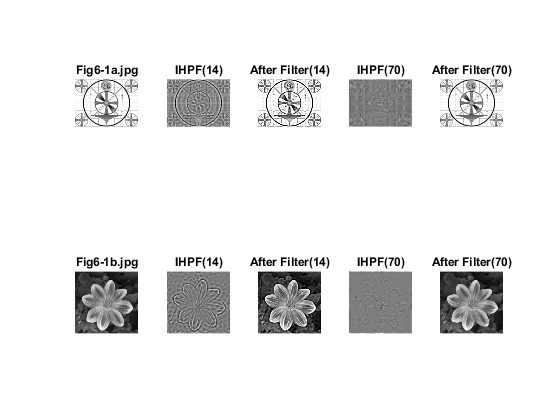


Figure - Images were applied to an ideal high-pass filter with two different distance from 0 arguments, varying the degree and intensity of the filter. As you can see from the images, there is artifacting rings shown around the changes between the bright and dark.

# Second-Order Butterworth Filter

imageA = im2double(imread("Fig6-1a.jpg"));

imageB = im2double(imread("Fig6-1b.jpg"));

D0\_1 =76;

D0\_2 =23;

A\_BHPF\_1 = bhpf(imageA, D0\_1);

A\_BHPF\_2 = bhpf(imageA, D0\_2);

B\_BHPF\_1 = bhpf(imageB, D0\_1);

B\_BHPF\_2 = bhpf(imageB, D0\_2);

A\_HP\_1 = imageA + A\_BHPF\_1;

A\_HP\_2 = imageA + A\_BHPF\_2;

B\_HP\_1 = imageB + B\_BHPF\_1;

B\_HP\_2 = imageB + B\_BHPF\_2;

figure();

% Image A

subplot(2,5,1)

imshow(imageA);

title("Fig6-1a.jpg");

subplot(2,5,2)

imshow(A\_BHPF\_1, []);

title("BHPF(" + D0\_1 + ")");

subplot(2,5,3)

imshow(A\_HP\_1);

title("After Filter(" + D0\_1 + ")");

subplot(2,5,4)

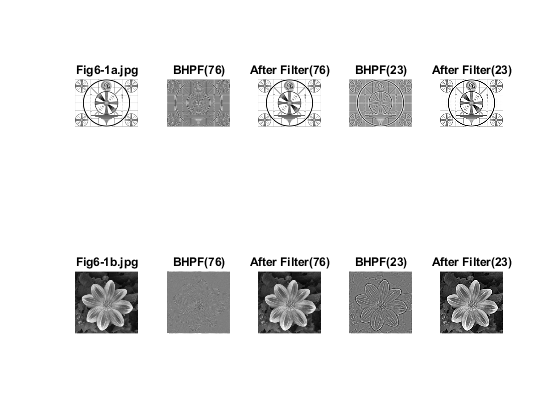
imshow(A\_BHPF\_2, []);

Figure - Images were applied to a second order Butterworth high-pass filter with two different distance from 0 arguments, varying the degree and intensity of the filter. As you can see from the images, the artifacting seen in the ideal filters are gone.

title("BHPF(" + D0\_2 + ")");

subplot(2,5,5)

imshow(A\_HP\_2);

title("After Filter(" + D0\_2 + ")");

% Image B

subplot(2,5,6)

imshow(imageB);

title("Fig6-1b.jpg");

subplot(2,5,7)

imshow(B\_BHPF\_1, []);

title("BHPF(" + D0\_1 + ")");

subplot(2,5,8)

imshow(B\_HP\_1);

title("After Filter(" + D0\_1 + ")");

subplot(2,5,9)

imshow(B\_BHPF\_2, []);

title("BHPF(" + D0\_2 + ")");

subplot(2,5,10)

imshow(B\_HP\_2);

title("After Filter(" + D0\_2 + ")");

# Gaussian Filter

imageA = im2double(imread("Fig6-1a.jpg"));

imageB = im2double(imread("Fig6-1b.jpg"));

D0\_1 =12;

D0\_2 =90;

A\_GHPF\_1 = ghpf(imageA, D0\_1);

A\_GHPF\_2 = ghpf(imageA, D0\_2);

B\_GHPF\_1 = ghpf(imageB, D0\_1);

B\_GHPF\_2 = ghpf(imageB, D0\_2);

A\_HP\_1 = imageA + A\_GHPF\_1;

A\_HP\_2 = imageA + A\_GHPF\_2;

B\_HP\_1 = imageB + B\_GHPF\_1;

B\_HP\_2 = imageB + B\_GHPF\_2;

figure();

% Image A

subplot(2,5,1)

imshow(imageA);

title("Fig6-1a.jpg");

subplot(2,5,2)

imshow(A\_GHPF\_1, []);

title("GHPF(" + D0\_1 + ")");

subplot(2,5,3)

imshow(A\_HP\_1);

title("After Filter(" + D0\_1 + ")");

subplot(2,5,4)

imshow(A\_GHPF\_2, []);

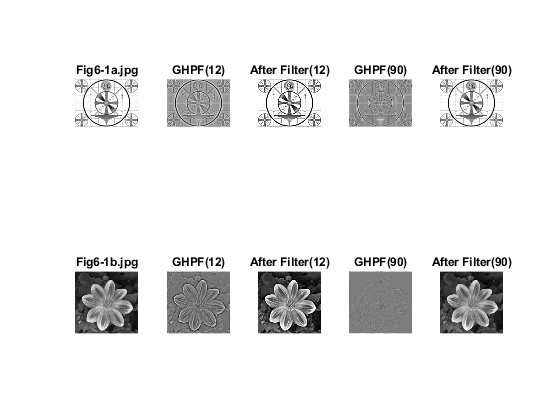
title("GHPF(" + D0\_2 + ")");

Figure - Images were applied to a gaussian high-pass filter with two different distance from 0 arguments, varying the degree and intensity of the filter. As you can see from the images, the artifacting seen in the ideal filters are gone.

subplot(2,5,5)

imshow(A\_HP\_2);

title("After Filter(" + D0\_2 + ")");

% Image B

subplot(2,5,6)

imshow(imageB);

title("Fig6-1b.jpg");

subplot(2,5,7)

imshow(B\_GHPF\_1, []);

title("GHPF(" + D0\_1 + ")");

subplot(2,5,8)

imshow(B\_HP\_1);

title("After Filter(" + D0\_1 + ")");

subplot(2,5,9)

imshow(B\_GHPF\_2, []);

title("GHPF(" + D0\_2 + ")");

subplot(2,5,10)

imshow(B\_HP\_2);

title("After Filter(" + D0\_2 + ")");

2. Band reject Filtering

# Apply a Band reject Filter to Fig5-2a.tif, Fig5-2b.tif. Fig6-2c.jpg

clear

imageA = imread('Fig6-2a.tif');

imageB = imread('Fig6-2b.tif');

imageC = imread("Fig6-2c.jpg");

[AS, AcS, AeS] = cceFourierSpec(imageA);

[BS, BcS, BeS] = cceFourierSpec(imageB);

[CS, CcS, CeS] = cceFourierSpec(imageC);

A\_C0 = 80;

A\_W = 1000;

A\_GBRF = gbrf(imageA, A\_C0, A\_W);

[AfS, AfcS, AfeS] = cceFourierSpec(A\_GBRF);

B\_C0 = 25;

B\_W = 3;

B\_GBRF = gbrf(imageB, B\_C0, B\_W);

[BfS, BfcS, BfeS] = cceFourierSpec(B\_GBRF);

C\_C0 = 165;

C\_W = 6.9;

C\_GBRF = gbrf(imageC, C\_C0, C\_W);

[CfS, CfcS, CfeS] = cceFourierSpec(C\_GBRF);

figure();

% Original Images

subplot(4,3,1)

imshow(imageA);

title("Fig6-2a.tif");

subplot(4,3,2)

imshow(imageB);

title("Fig6-2b.tif");

subplot(4,3,3)

imshow(imageC);

title("Fig6-2c.jpg");

% Enhanced Centered Spectrums

subplot(4,3,4)

imshow(AeS, []);

title("Log Centered Spectrum");

subplot(4,3,5)

imshow(BeS, []);

title("Log Centered Spectrum");

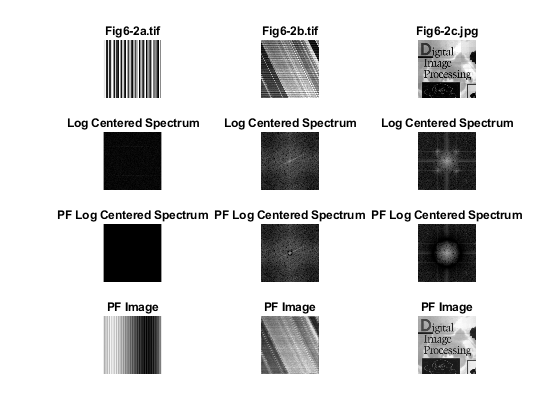
subplot(4,3,6)

imshow(CeS, []);

title("Log Centered Spectrum");

% Post Filter Centered Spectrums

subplot(4,3,7)

imshow(AfeS, []);

title("PF Log Centered Spectrum");

subplot(4,3,8)

imshow(BfeS, []);

title("PF Log Centered Spectrum");

subplot(4,3,9)

imshow(CfeS, []);

title("PF Log Centered Spectrum");

% Post Filter Images

subplot(4,3,10)

imshow(A\_GBRF, []);

title("PF Image")

subplot(4,3,11)

imshow(B\_GBRF, []);

title("PF Image")

subplot(4,3,12)

imshow(C\_GBRF, []);

title("PF Image")

Explanation: Each of the images were applied with a Gaussian Band Reject filter with unique parameters adjusted per image to eliminated various errors. In the first image, the underlying gradient was revealed by eliminating all unnecessary frequencies. In the second image, the horizontal lines were eliminated by a similar band reject filter, but in this case the horizontal lines are close to the center of the frequency spectrum. In the third image, the crosshatch pattern was eliminated by eliminating the 4 loci located around the center area on the spectrum.

3. Filtering

# Filter and Threshold Fig6-3.tif to optimize image clarity

clear

image = im2double(imread('Fig6-3.tif'));

[S, cS, eS] = cceFourierSpec(image);

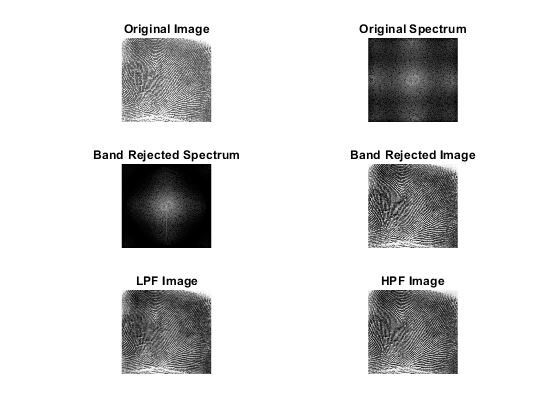
C0 = 481;

W = 25.3;

GBRF = gbrf(image, C0, W);

[FS, FcS, FeS] = cceFourierSpec(GBRF);

D0\_L = 61;

LP = glpf(GBRF, D0\_L);

D0\_H = 40;

HPF = ghpf(LP, D0\_H);

HP = LP + HPF;

figure();

subplot(3,2,1)

imshow(image);

title("Original Image");

subplot(3,2,2)

imshow(eS, []);

title("Original Spectrum");

subplot(3,2,3)

imshow(FeS, []);

title("Band Rejected Spectrum");

subplot(3,2,4)

imshow(GBRF, []);

title("Band Rejected Image");

subplot(3,2,5)

imshow(LP, []);

title("LPF Image");

subplot(3,2,6)

imshow(HP, []);

title("HPF Image");

Explanation: Multiple filters were applied to this fingerprint image to improve the clarity of the pattern. First a band reject filter was applied to eliminate unnecessary and noisy high frequencies. Then a low pass filter was applied to join potentially disjoint lines. Then a high pass filter was applied to the low pass filtered image to show the edges more.

# Appendix (Functions Used)

function filtered = ihpf(imag, D0)

% Description: Perform an Ideal High-Pass Filter to an image with the given sigma

% Initial Variables

[M,N] = size(imag);

F = fft2(im2double(imag));

% Mesh grid generation

u = 0:(M-1);

v = 0:(N-1);

idx = find(u>M/2);

u(idx)= u(idx)-M;

idy = find(v>N/2);

v(idy)= v(idy)-N;

[U,V] = meshgrid(v,u);

D = sqrt(U.^2+V.^2);

% Ideal Filter

H = double(D>D0);

% Apply Filter

G = F.\*H;

filtered = real(ifft2(G));

end

function filtered = bhpf(imag, D0)

% Description: Perform an Butterworth High-Pass Filter to an image with the given sigma

% Initial variables

[M,N] = size(imag);

F = fft2(im2double(imag));

% Mesh grid generation

u = 0:(M-1);

v = 0:(N-1);

idx = find(u>M/2);

u(idx)= u(idx)-M;

idy = find(v>N/2);

v(idy)= v(idy)-N;

[U,V] = meshgrid(v,u);

D = sqrt(U.^2+V.^2);

% Butterworth Filter: default n value

n=2;

H=1./(1+(D0./D).^(2\*n));

% Apply Filter

G = F.\*H;

filtered = real(ifft2(G));

end

function filtered = ghpf(imag, D0)

% Description: Perform an Gaussian High-Pass Filter to an image with the given sigma

% Initial variables

[M,N] = size(imag);

F = fft2(im2double(imag));

% Mesh grid generation

u = 0:(M-1);

v = 0:(N-1);

idx = find(u>M/2);

u(idx)= u(idx)-M;

idy = find(v>N/2);

v(idy)= v(idy)-N;

[U,V] = meshgrid(v,u);

D = sqrt(U.^2+V.^2);

% Gaussian Filter

H = 1-exp(-(D.^2)./(2\*(D0^2)));

% Apply Filter

G = F.\*H;

filtered = real(ifft2(G));

end

function filtered = gbrf(imag, C0, W)

% Description: Perform an Gaussian Band Reject Filter to an image with the given

% Center and Width of the band

% Initial variables

[M,N] = size(imag);

F = fft2(im2double(imag));

% Mesh grid generation

u = 0:(M-1);

v = 0:(N-1);

idx = find(u>M/2);

u(idx)= u(idx)-M;

idy = find(v>N/2);

v(idy)= v(idy)-N;

[U,V] = meshgrid(v,u);

D = sqrt(U.^2+V.^2);

% Gaussian Filter

H = 1-exp(-(((D-C0).^2)./(D.\*W^2)));

% Apply Filter

G = F.\*H;

filtered = real(ifft2(G));

end

function [S,cS,eS] = cceFourierSpec(imag)

% Description: Compute, Center, and Enhance a Fourier Spectrum of an Image

% Base FFT

F = fft2(im2double(imag));

% Spectrum

S = abs(F);

% Centered Spectrum

cF = fftshift(F);

cS = abs(cF);

% Enhanced Spectrum

eS = log(1+abs(cF));

end

function filtered = blpf(imag, D0)

% Description: Perform an Butterworth Low-Pass Filter to an image with the given sigma

% Initial variables

[M,N] = size(imag);

F = fft2(im2double(imag));

% Mesh grid generation

u = 0:(M-1);

v = 0:(N-1);

idx = find(u>M/2);

u(idx)= u(idx)-M;

idy = find(v>N/2);

v(idy)= v(idy)-N;

[U,V] = meshgrid(v,u);

D = sqrt(U.^2+V.^2);

% Butterworth Filter: default n value

n=2;

H=1./(1+(D./D0).^(2\*n));

% Apply Filter

G = F.\*H;

filtered = real(ifft2(G));

end

function filtered = glpf(imag, D0)

% Description: Perform an Gaussian Low-Pass Filter to an image with the given sigma

% Initial variables

[M,N] = size(imag);

F = fft2(im2double(imag));

% Mesh grid generation

u = 0:(M-1);

v = 0:(N-1);

idx = find(u>M/2);

u(idx)= u(idx)-M;

idy = find(v>N/2);

v(idy)= v(idy)-N;

[U,V] = meshgrid(v,u);

D = sqrt(U.^2+V.^2);

% Gaussian Filter

H = exp(-(D.^2)./(2\*(D0^2)));

% Apply Filter

G = F.\*H;

filtered = real(ifft2(G));

end