Step-by-step Examples for Selected Techniques
Prepared by Dr. Xiaojun Qi
Oct. 15, 2024

Technique 1: Histogram Equalization (4 steps)

1. Obtain the histogram H(k)
2. Compute the cumulative normalized histogram T(k)
3. Compute the transformed intensity by: $g_k = (L-1) * T(k)$, where L is the maximum gray level of the new processed image
4. Scan the image and set the pixel with the intensity k to g_k.

Given a 3-bit Grayscale image, whose gray level intensity is in the range of [0, 7]

1	1	7	6
1	3	4	3
6	6	7	1
2	5	5	3

Performing histogram equalization technique on the original image to obtain a 8-bit grayscale image, whose gray level intensity is in the range of [0, 255].

1. Histogram	Normalized Histogram	2. Cumulative Normalized histogram	3. G(k)	4. Mapping (transform function)
H(0)=0	P(0)=0	T(0)=0	G(0)=0	0→0
H(1)=4	P(1)=1/4	T(1)=1/4	G(1)=64	1→64
H(2)=1	P(2)=1/16	T(2)=5/16	G(2)=5/16*255=80	2→80
H(3)=3	P(3)=3/16	T(3)=8/16	G(3)=128	3→128
H(4)=1	P(4)=1/16	T(4)=9/16	G(4)=9/16*255=143	4→143
H(5)=2	P(5)=2/16	T(5)=11/16	G(5)=11/16*255=175	5→175
H(6)=3	P(6)=3/16	T(6)=14/16	G(6)=14/16*255=223	6→223
H(7)=2	P(7)=2/16	T(7)=1	G(7)=255	7→255

Histogram equalization transformed image will be:

64	64	255	223
64	128	143	128
223	223	255	64
80	175	175	128

Technique 2: Fourier Transform

In a sine wave C*Sin(Ax + B), A represents frequency, B represents phase, and C represents magnitude.

The Nyquist Theorem states the following: The **nyquist sampling rate** is two times the highest **frequency** of the input signal.

Spatial Domain

f(0)	f(1)	f(2)	f(3)
2	3	4	4

Frequency Domain

F(0) 0 frequency	F(1) 1 frequency	F(2)	F(3)
13/4	(-2+i)/4	-1/4	(-2-i)/4

Spatial domain (x, y)

x

f(0,0) ; 3	f(1, 0) ; 6	f(2, 0) ; 9	f(3, 0) ; 10
f(0, 1) ; 2	f(1, 1) ; 4	f(2, 1) ; 9	f(3, 1) ; 0
f(0, 2) ; 10	f(1, 2) ; 3	f(2, 2) ; 9	f(3, 2) ; 8
f(0, 3) ; 20	f(1, 3) ; 4	f(2, 3) ; 8	f(3, 3) ; 14

y

Frequency domain (u, v)

u

F(0,0) ; 1.19	F(1, 0) 0 + 0.15i	F(2, 0) ; 0.21	F(3, 0) ; 0 - 0.15i
F(0, 1) ; -0.02 + 0.31i	F(1, 1); -0.21 + 0.18i	F(2, 1) ; -0.12 + 0.03i	F(3, 1) ; 0.07 + 0.20i
F(0, 2) ; -0.03	F(1, 2) ; -0.10 + 0.03i	F(2, 2) ; -0.13 (Center)	F(3, 2) ; -0.10 - 0.03i
F(0, 3) ; -0.02 - 0.31i	F(1, 3) ; 0.07 - 0.20i	F(2, 3) ; -0.12 - 0.03i	F(3, 3); -0.21 - 0.18i

v

Conclusion:
For a given image of size n*n, calling **fft2** function can convert the image into a frequency image of the same size n*n, where the upper-left corner holds the information of the zero frequency.

The first quadrant of the frequency image contains the valid sine wave information, where frequency is indicated in the location in the frequency image, and magnitude and phase are computed using the equations listed on slides 23 Ch3.2.DIPBasicFreq.pdf (page 4).

The frequency image has complex conjugate pairs symmetrically around the center.

Technique 3: Filtering Technique in the Fourier Transform Domain

We will use the image coordinate, e.g., (row, column) representation, to refer to each pixel in both spatial and frequency domains.

Below are the key steps to perform filtering operations in the frequency domain.

Given an image A in the spatial domain (x, y)

y (column)

f(0,0) ; 3	f(0, 1) ; 6	f(0, 2) ; 9	f(0, 3) ; 10
f(1, 0) ; 2	f(1, 1) ; 4	f(1, 2) ; 9	f(1, 3) ; 0
f(2, 0) ; 10	f(2, 1) ; 3	f(2, 2) ; 9	f(2, 3) ; 8
f(3, 0) ; 20	f(3, 1) ; 4	f(3, 2) ; 8	f(3, 3) ; 14

x (row)

Step 1: Calling **fft2(A)** function converts the image A to its image B in the frequency domain (u, v)

v (Column)

F(0, 0) ; 1.19	F(0, 1) 0 + 0.15i	F(0, 2) ; 0.21	F(0, 3) ; 0 - 0.15i
F(1, 0) ; -0.02 + 0.31i	F(1, 1); -0.21 + 0.18i	F(1, 2) ; -0.12 + 0.03i	F(1, 3) ; 0.07 + 0.20i
F(2, 0) ; -0.03	F(2, 1) ; -0.10 + 0.03i	F(2, 2) ; -0.13 (Center)	F(2, 3) ; -0.10 - 0.03i
F(3, 0) ; -0.02 - 0.31i	F(3, 1) ; 0.07 - 0.20i	F(3, 2) ; -0.12 - 0.03i	F(3, 3); -0.21 - 0.18i

u (row)

Step 2: Calling **fftshift(B)** function generates C, which has the lowest frequency at the center in the frequency domain (u, v)

v

F(2, 2) ; -0.13 Pos(1, 1)	F(2, 3) ; -0.10 - 0.03i	F(2, 0) ; -0.03	F(2, 1) ; -0.10 + 0.03i
F(3, 2) ; -0.12 - 0.03i	F(3, 3); -0.21 - 0.18i	F(3, 0) ; -0.02 - 0.31i	F(3, 1) ; 0.07 - 0.20i
F(0, 2) ; 0.21	F(0, 3) ; 0 - 0.15i	F(0, 0) ; 1.19 Center pos(3, 3)	F(0, 1) 0 + 0.15i Pos(3, 4)
F(1, 2) ; -0.12 + 0.03i	F(1, 3) ; 0.07 + 0.20i	F(1, 0) ; -0.02 + 0.31i	F(1, 1); -0.21 + 0.18i

Step 3: Design a filter H, which has the same dimension as the original image A, using equations (slides 54, 59, and 63)
For example:
function ILPF = DesignILPF(H, W, D0), where H is the height of the filter, W is the width of the filter, D0 is the cut-off frequency, and ILPF is the ideal low-pass filter.

Step 4: Perform **elementwise multiplication** between C (result obtained from step 2) and H (result obtained from step 3) to generate filtered image in the frequency domain.
Result1 = C **.*** H ;

Step 5: Go back to spatial domain to generate filtered image.
filteredA = **ifft2**(**ifftshift**(Result1))

Conclusions:

1) Filter in the frequency domain has the same dimension as the original image.
2) **Element-wise multiplication is performed in the frequency domain.** This operation is equivalent to the convolution operation in the original spatial domain.

Technique 4: Wavelet Transform

Given a pair of data (a b), compute C = (a+b)/2 ; D = (a-b)/2

Original Data	**a = 77**	**b = 80**	**0**	**255**	**1**	**100**	**150**	**0**
	(a+b)/2 =157/2= 78.5	(a-b) = -3/2 = -1.5	255/2= 127.5	-127.5	101/2= 50.5	-99/2= -49.5	75	75
1st-level Wavelet decomposition	**78.5 Approximation Coefficients**	**127.5**	**50.5**	**75**	**-1.5 1st-level Detail Coefficients**	**-127.5**	**-49.5**	**75**
2nd-level Wavelet decomposition	(78.5+127.5)/2 =103	(50.5+75)/2 =62.75	(78.5-127.5)/2 =-24.5 2nd-level detail coef.	(50.5-75)/2 = -12.25 Change To 0	-1.5	-127.5	-49.5	75
3rd-level Wavelet decomposition	(103+62.75)/2 Approximation Coef.	(103-62.75)/2 3rd-level detail coef.	-24.5	-12.25	-1.5	-127.5	-49.5	75

Filters used in the above example:
Lowpass filter (LPF): [0 0.5 0.5] or [0.5 0.5] → Obtain the approximation coefficients
Highpass filter (HPF): [0 0.5 -0.5] or [0.5 -0.5] → Obtain the detail coefficients

Haar Wavelet:
LPF: [1/sqrt(2) 1/sqrt(2)]
HPF: [1/sqrt(2) -1/sqrt(2)]

One implementation:

Step 1: Set the decomposition mode to ensure that the size after each decomposition is reduced by half

dwtmode('per') ;

im = imread('Lena.jpg');

Step 2: call dwt2 to perform one level Discrete Wavelet Transformation (i.e., decomposition) on a 2D image

[CA1, CH1, CV1, CD1] = dwt2(im, 'db2') ; % db2 means filters with 4 values

[CA2, CH2, CV2, CD2] = dwt2(CA1, 'db2') ; % db2 means filters with 4 values

[CA3, CH3, CV3, CD3] = dwt2(CA2, 'db2') ; % db2 means filters with 4 values

Step 3: Perform some desired operations

CH3 = 0 ; % Any processing operations

CD3(1:2, 1:2) = 5 ;

Step 4: call idwt2 to perform Inverse Discrete Wavelet Transformation (i.e., reconstruction) on the decomposed image to reconstruct the image

newCA2 = idwt2(CA3, CH3, CV3, CD3, 'db2');

newCA1 = idwt2(newCA2, CH2, CV2, CD2, 'db2');

newIm = idwt2(newCA1, CH1, CV1, CD1, 'db2');

Another Implementation:

Step 1: Set the decomposition mode to ensure that the size after each decomposition is reduced by half

dwtmode('per') ;

im = imread('Lena.jpg');

Step 2: call wavedec2 to perform multi-level Discrete Wavelet Transformation (i.e., decomposition) on a 2D image

[c1, s1] = wavedec2(im, 3, 'db2') ; % Perform 3-level decomposition

Step 3: Perform some desired operations

c1(30:40) = -10 ; % Any processing operations

c1(1:10) = 0 ;

Step 4: call waverec2 to perform multi-level Inverse Wavelet Transformation (i.e., reconstruction) on the decomposed image to reconstruct the image

newIm = waverec2(c1, s1, 'db2') ;