Image Processing

Frequency Domain Processing (Part I)

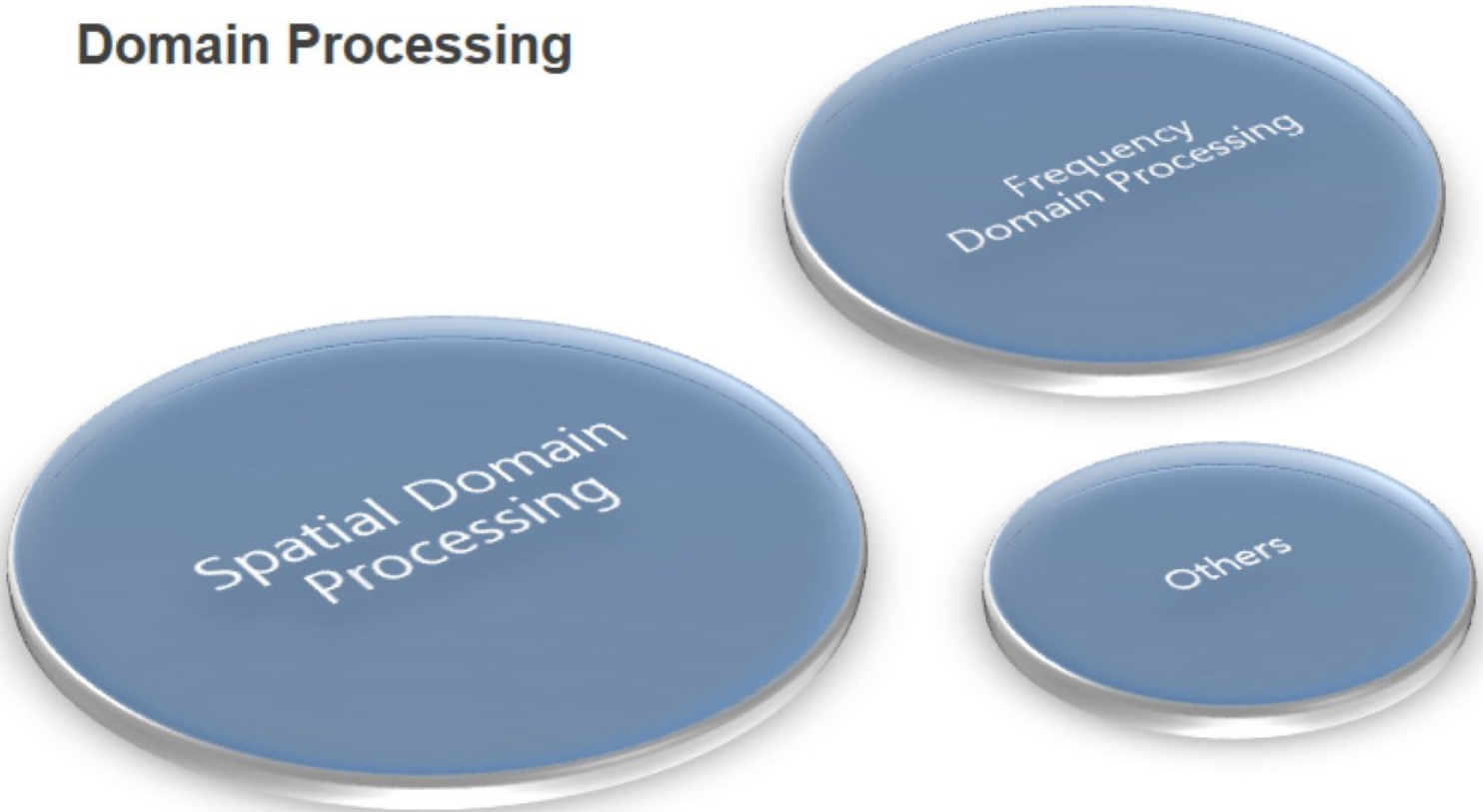
Pattern Recognition and Image Processing Laboratory (Since 2012)

Introduction

Transformation

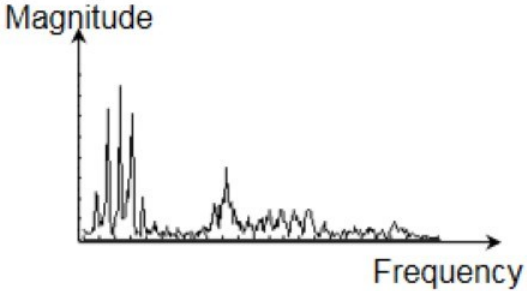
Introduction

Domain Processing

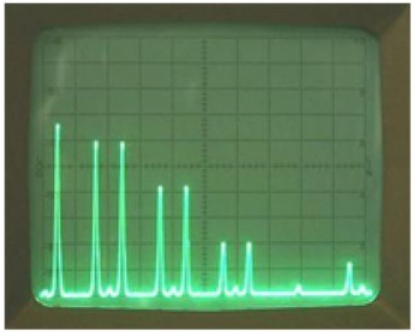


Introduction

What is a Transform?



Time Domain → Transform → Frequency Domain



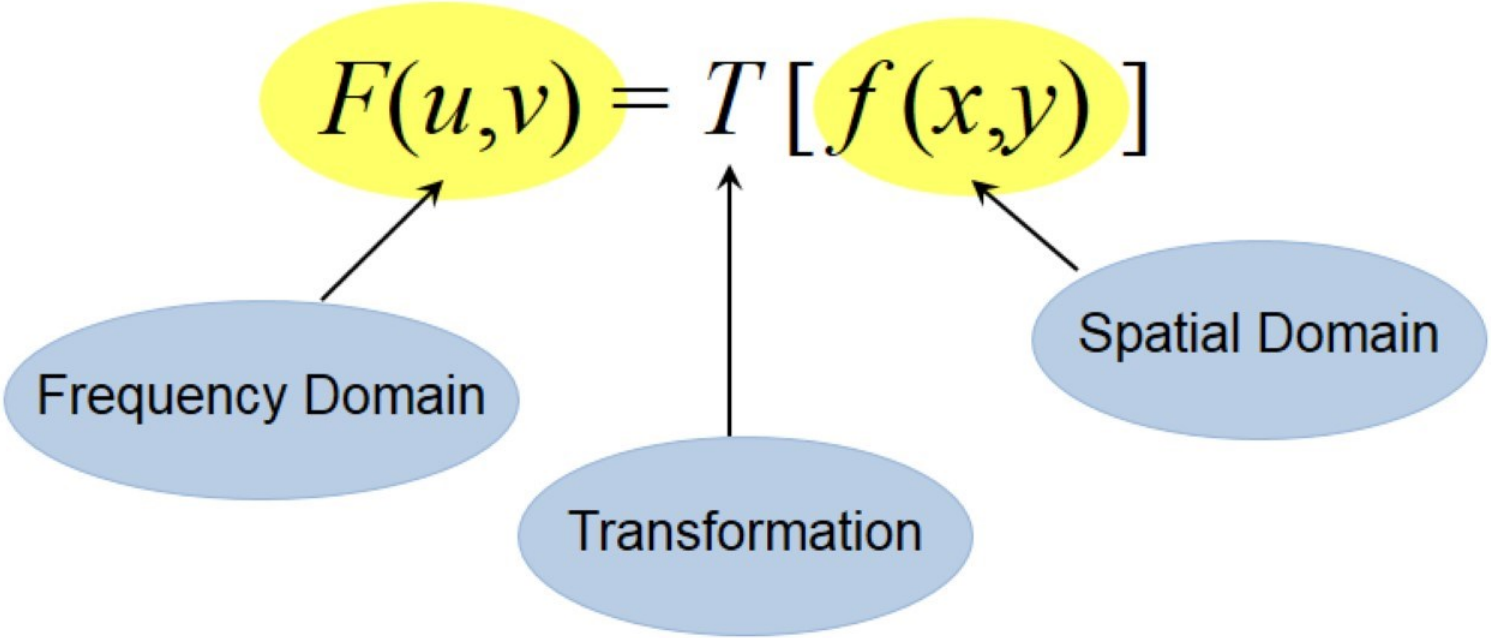
Introduction

Types of Transforms

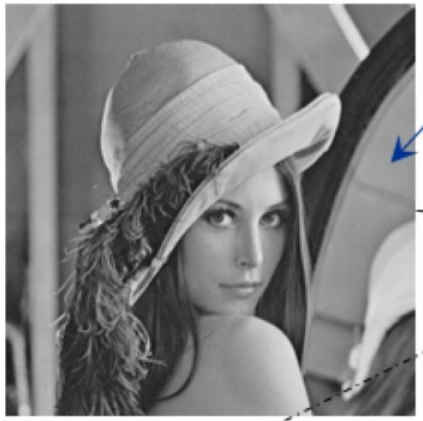
- Fourier Transform
- Hanamard Transform
- KLT Transform
- Discrete Cosine Transform
- Wavelet Transform
- ...

2D Discrete Fourier Transform

A frequency domain processing is denoted by the expression.

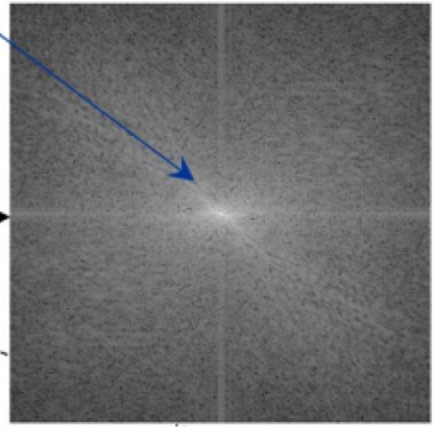


2D Discrete Fourier Transform



$f(x,y)$: pixels

$F(u,v)$: DFT coefficients

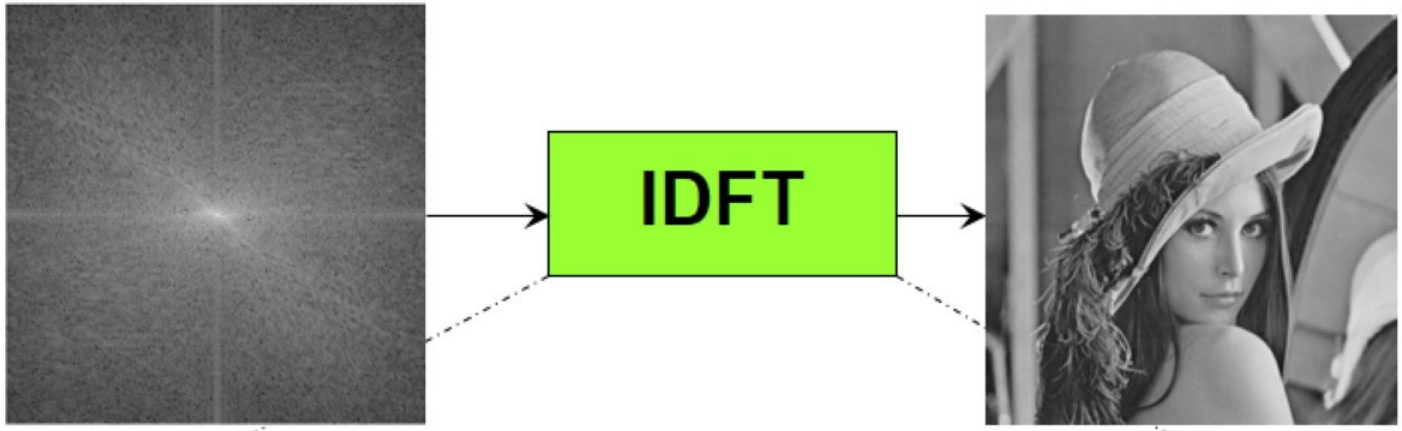


DFT

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)};$$

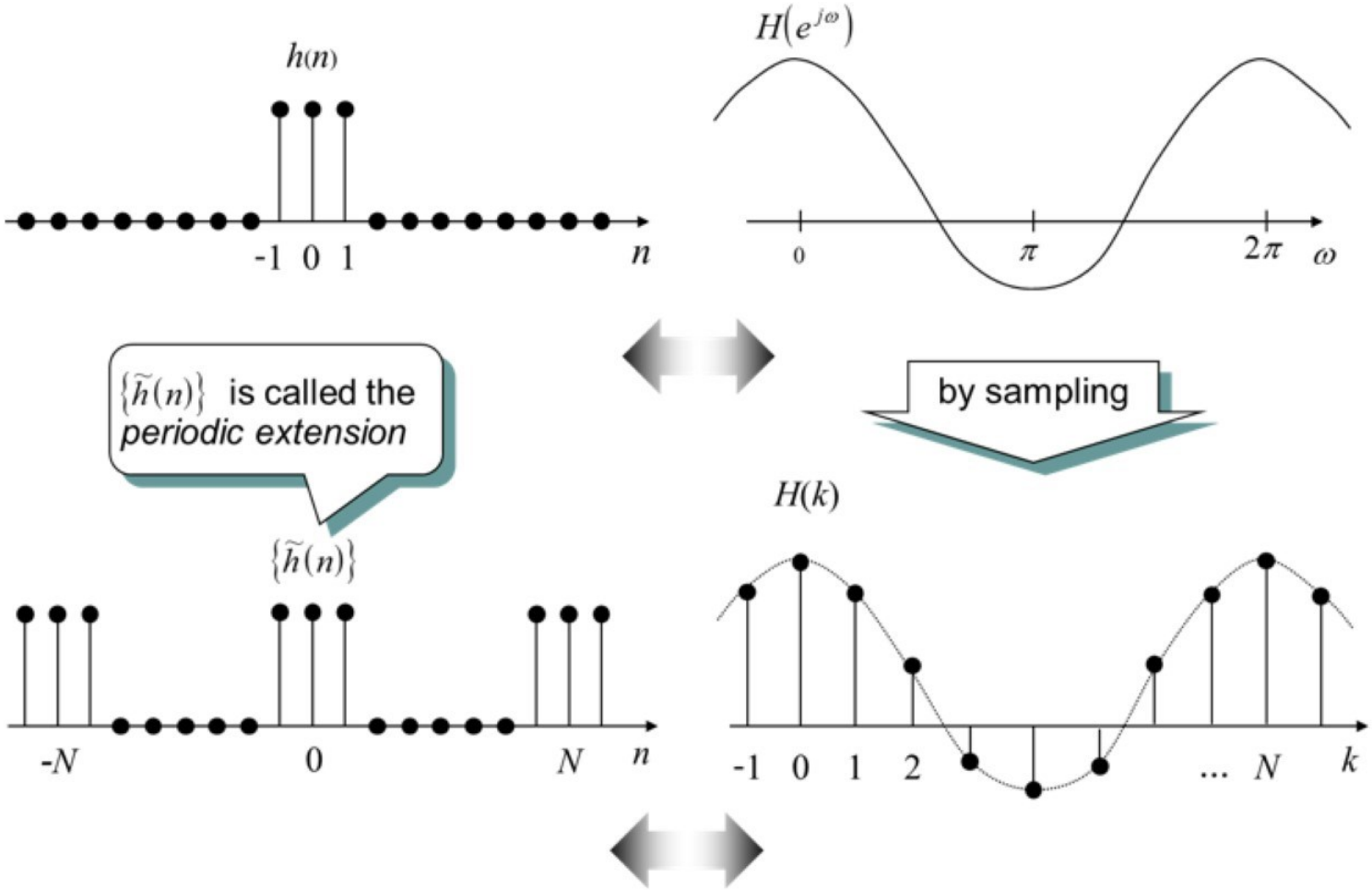
for $u = 0, 1, 2, \ldots, M\text{-}1$ and
$v = 0, 1, 2, \ldots, N\text{-}1.$

2D Discrete Fourier Transform



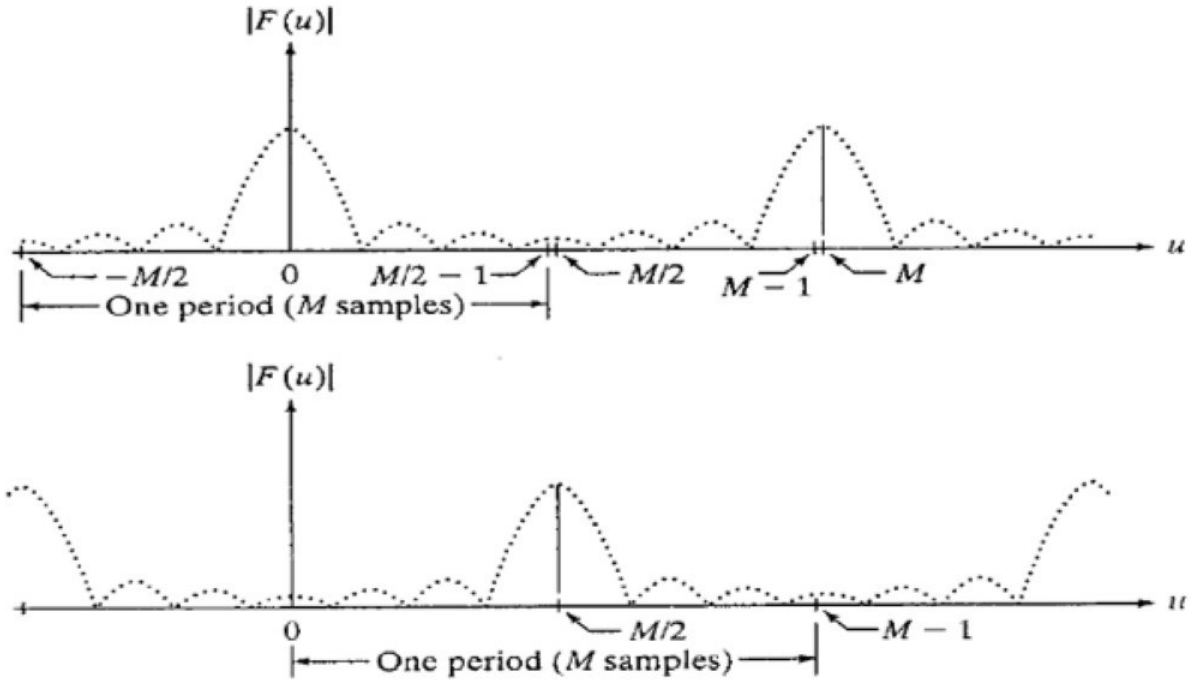
$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)};$$

for $x = 0, 1, 2, \ldots, M\text{-}1$ and
$y = 0, 1, 2, \ldots, N\text{-}1.$


h(n)
-1 0 1
n
H(e^{jω})
0
π
2π
ω
by sampling
{h̃(n)} is called the periodic extension
{h̃(n)}
-N
0
N
n
H(k)
-1 0 1 2
... N
k

2D Discrete Fourier Transform

Periodicity property of DFT: 1-D case



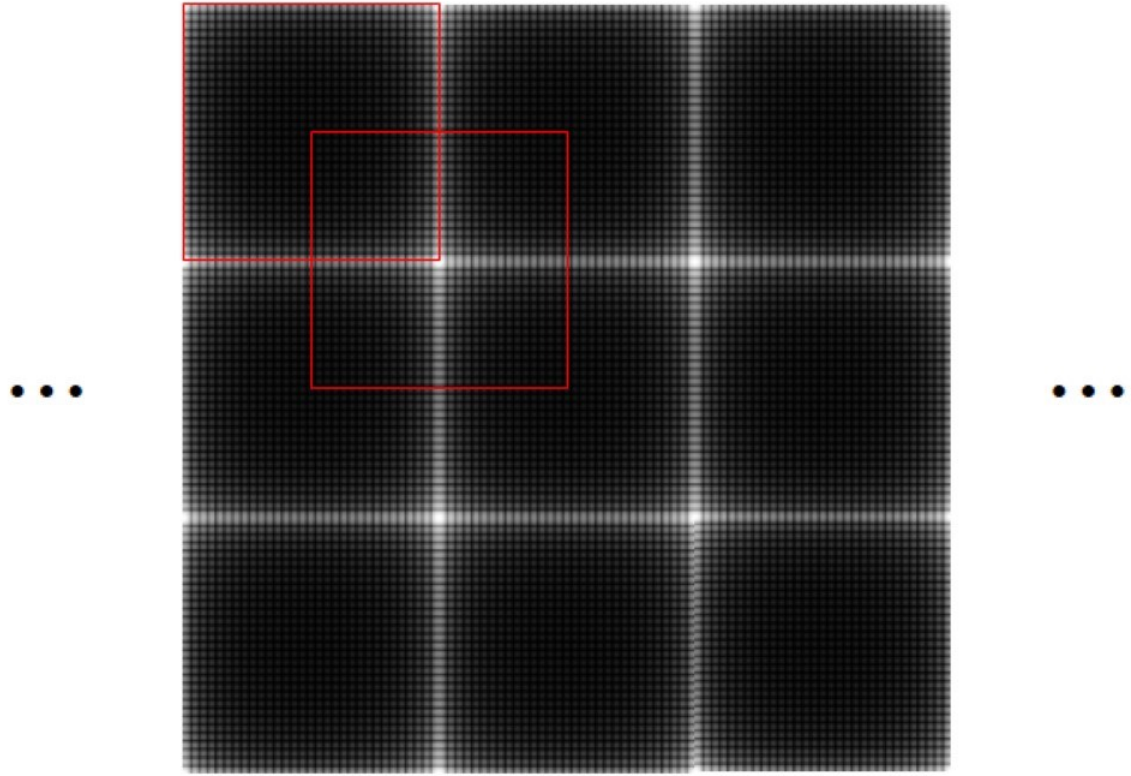
2D Discrete Fourier Transform

Periodicity property of DFT: 1-D case

>> Additional Example % See demonstration

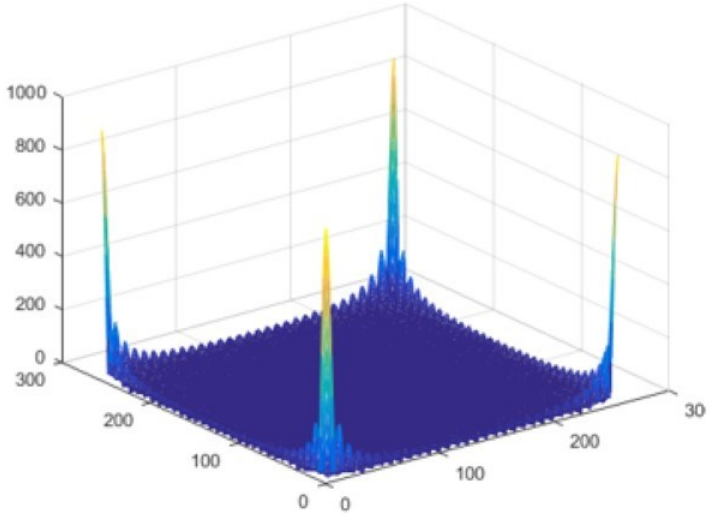
2D Discrete Fourier Transform

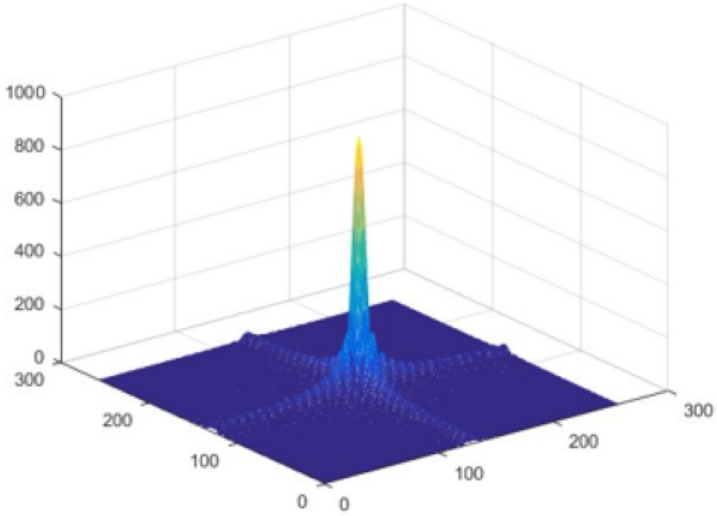
Periodicity property of DFT: 2-D case



2D Discrete Fourier Transform

Computing and Visualizing the 2-D DFT in MATLAB





2D Discrete Fourier Transform

Computing and Visualizing the 2-D DFT in MATLAB

>> ex4_01 % See demonstration

Filtering in the Frequency Domain

Fundamental Concepts

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

$$f(x,y)h(x,y) \Leftrightarrow F(u,v) * H(u,v)$$

Filtering in the Frequency Domain

```
>> ex4_02  % See demonstration
```


The end of
part I

```
%% --------------- Additional Example ----------------------
clear all
close all

f1 = [1 1 1]
F1 = fft(f1);
figure, stem(abs(F1));

f10 = [1 1 1 0 0 0 0 0 0 0]
F10 = fft(f10);
figure, stem(abs(F10));

f20 = [1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
F20 = fft(f20);
figure, stem(abs(F20));

% -------------------- simulation of ideal FT ----------------------
w = 0:2*pi/40:2*pi;
y = exp(-j.*2.*w).*(1+2.*cos(w));
figure, plot(w,abs(y))
```

```
ex4_01.m
%% ------------------------------------

clear all;
close all;

f = zeros(256,256);
f(128-15:128+15, 128-15:128+15) = 1;
figure(1);
imshow(f);

F = fft2(double(f));
figure(2);
imshow(abs(F), []);

Fc = fftshift(F);
figure(3);
imshow(abs(Fc), []);

S2 = log(1+abs(Fc));
figure(4);
imshow(abs(S2), []);
figure(5);
mesh(abs(F));
```

ex4_02.m

```
%% ------------------------------------

% An example of zero padding

up = ones(128,256);
low = zeros(128,256);
bw = [up; low];
figure(1); imshow(bw);

[M, N] = size(bw);
BW = fft2(bw);
sig = 10;
H = lpfilter('gaussian', M, N, sig);
G = H.*BW;
g = real(ifft2(G));
figure(2); imshow(g, []);

PQ = paddedsize(size(bw));
BW1 = fft2(bw, PQ(1), PQ(2));
sig = 10;
H1 = lpfilter('gaussian', PQ(1), PQ(2),
2*sig); % 2*sig, because the filter size

% is now twice the size of

% the filter used without padding.
G1 = H1.*BW1;
g1 = real(ifft2(G1));
figure(3); imshow(g1, []);

gc1 = g1(1:size(bw,1), 1:size(bw,2));
figure(4); imshow(gc1, []);
```

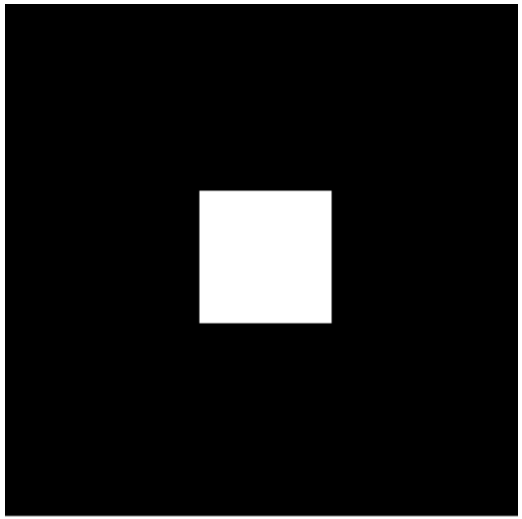
Image Processing

Workshop on Frequency Domain Processing (Part I)

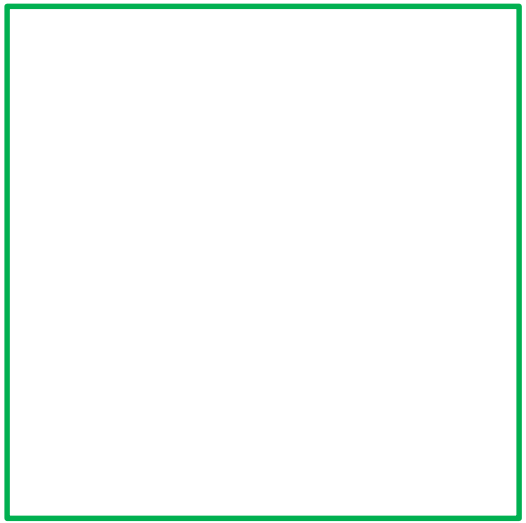
Pattern Recognition and Image Processing Laboratory (Since 2012)

Workshop on Frequency Domain Processing (Part I)

1. ใช้ฟังก์ชัน imread เพื่ออ่านภาพชื่อ SQ.png ดังภาพที่ (1.1) เข้ามาเก็บไว้ในตัวแปร f1 จากนั้นให้แปลงภาพ f1 ด้วยฟังก์ชัน fft2 แล้วแสดงผลลัพธ์ที่ได้ในภาพที่ (1.2)



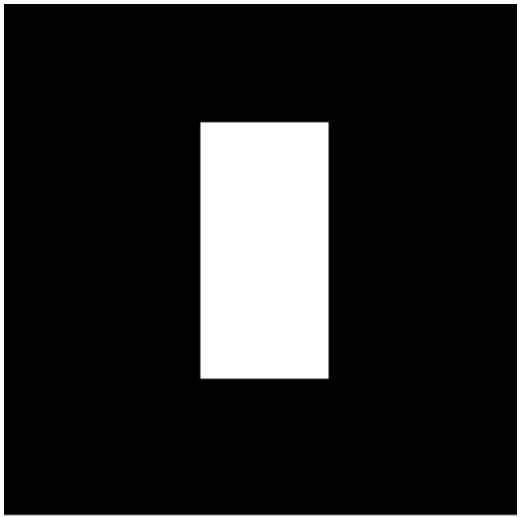
(1.1)



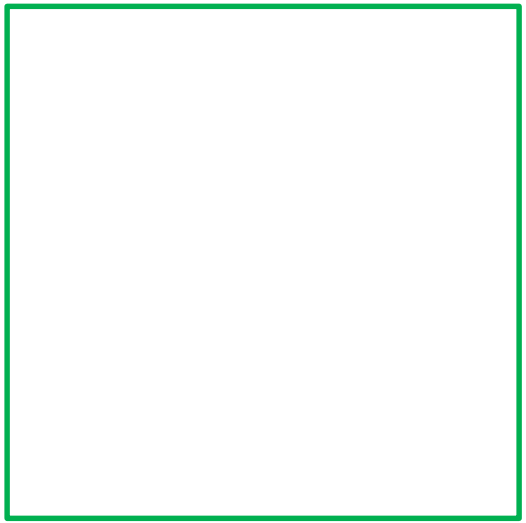
(1.2)

Workshop on Frequency Domain Processing (Part I)

2. ใช้ฟังก์ชัน imread เพื่ออ่านภาพชื่อ Reg_0.png ดังภาพที่ (2.1) เข้ามาเก็บไว้ใน ตัวแปร f2 จากนั้นให้แปลงภาพ f2 ด้วยฟังก์ชัน fft2 แล้วแสดงผลลัพธ์ที่ได้ในภาพที่ (2.2)



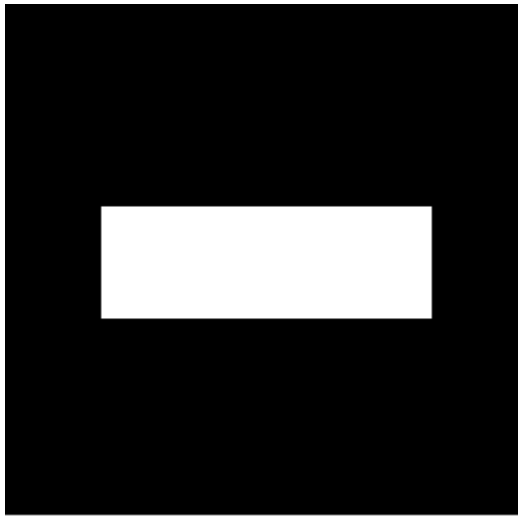
(2.1)



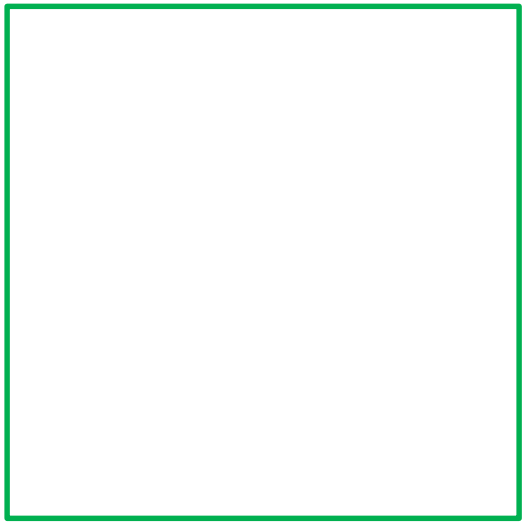
(2.2)

Workshop on Frequency Domain Processing (Part I)

3. ใช้ฟังก์ชัน imread เพื่ออ่านภาพชื่อ Reg_90.png ดังภาพที่ (3.1) เข้ามาเก็บไว้ใน ตัวแปร f3 จากนั้นให้แปลงภาพ f3 ด้วยฟังก์ชัน fft2 แล้วแสดงผลลัพธ์ที่ได้ในภาพที่ (3.2)



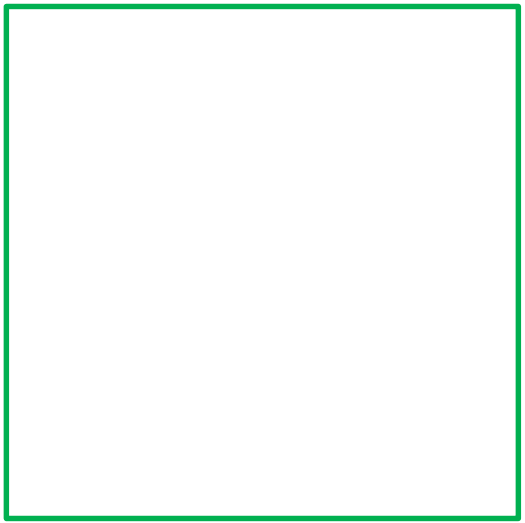
(3.1)



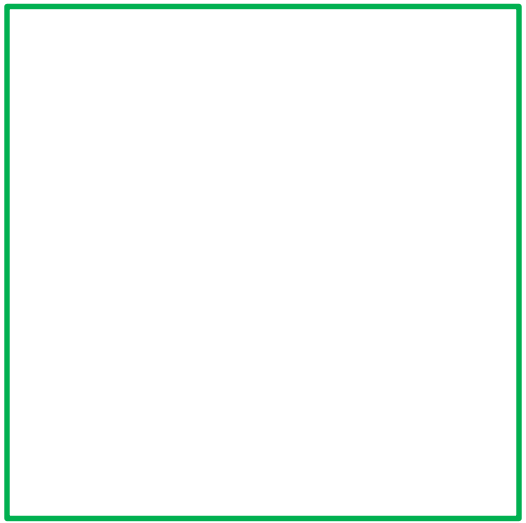
(3.2)

Workshop on Frequency Domain Processing (Part I)

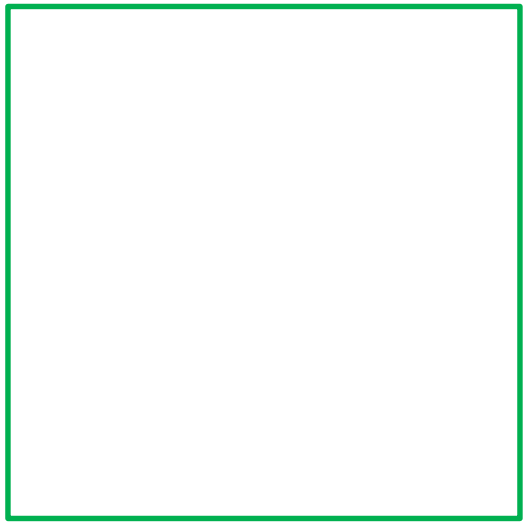
4. ให้เปรียบเทียบผลลัพธ์ที่ได้จากข้อ 1-3 เมื่อภาพที่ถูกแปลงด้วย fft2 มีคุณลักษณะที่เปลี่ยนไป



(1.2)



(2.2)



(3.2)

Workshop on Frequency Domain Processing (Part I)

5. ใช้ฟังก์ชัน imread เพื่ออ่านภาพชื่อ c0.png ถึง c5.png แล้วทำการแปลงภาพ ทั้ง 6 ภาพ ด้วยฟังก์ชัน fft2
6. ให้สังเกตผลลัพธ์ที่ได้จากข้อ 5 ว่ามีคุณลักษณะเช่นไร