Gamma-ray Imaging – PET, nuclear medicine, astronomy. แฝรังสีแล้วตรวจจับ
X-ray Imaging – CAT scans, angiography, industrial inspection. ดูการดูดซึม
Ultraviolet Imaging – microscopy, astronomy, laser studies.
Visible & Infrared – microscopy, remote sensing, Weather observation.
Microwave – Radar, Penetration through surfaces.
Radio Band – MRI (Magnetic Resonance Imaging) Medical.
Other Modalities – ultrasound, electron microscopy, computer-generated images.

Human Eye:
- Lens focuses light onto the retina. retina is where image actually form.
- Retina have 2 types of receptors
  - Cones (6-7M) → color + details (at the central of retina).
  - Rods (75-150M) → overall, sensitive to illumination, no color perception distribute all over the retina.

BTW, image on retina is up-side down, brain flip it

Electromagnetic Spectrum: visible light (400-700nm) is just one portion.

Image Acquisition Devices:
- Single sensor (requires motion), line sensor, array sensor (e.g., CCD cameras).
  - sensor is made from filter + sensing material (CMOS,CCD) + power
- Image = combination of illumination + reflectance. ภาพเกิดจากแสงสะท้อนของวัตถุ

Smoothing (Low-pass)
- Purpose: blur / ลด noise
- Methods:
  - Averaging filter (box filter) -> blur edge, box filter คือ weighted ที่มีแต่ 1
  - Weighted average (Gaussian mask) -> อย่าลืม coeff รวมมาหารด้วย
  - Median filter → ดีมากกับ impulse noise (salt & pepper)
- ผลลัพธ์ขอบเบลอ ส่วนใหญ่เพราะ Padding ด้วย 0

Sharpening (High-pass)
- Purpose: เน้นขอบ (edges), fine details

Image border:
- partial filter mask
- use 'padding' the image by adding rows and columns of 0's
- padding by replicating rows or columns.
- [perfectly filtered] limit excursions of the centre of the filter mask to than (n-1)/2 pixels from the border.

Laplacian | Laplacian + ori | Gx | Gy

Gx detects horizontal edges; Gy for vertical edges.

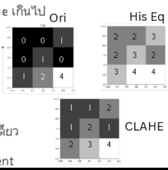
0	-1	0
-1	4	-1
0	-1	0

0	-1	0
-1	4+1	-1
0	-1	0

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

CLAHE (Contrast Limited Adaptive Histogram Equalization)
- Equalization แบบแบ่งย่อยเป็น **grids** (ตัดเป็นส่วนๆ แล้วแยกทำแต่ละส่วน)
- ใช้ clip limit ป้องกันไม่ให้ contrast extreme เกินไป
- Steps:
  1. แบ่งภาพเป็น grid
  2. คำนวณ histogram ของแต่ละ grid
  3. Clip ค่าเกิน limit
  4. ใช้ CDF mapping ใหม่
  5. Interpolation รวมกลับมาเป็นภาพเดียว



ใช้ใน medical imaging, low-light enhancement

Weighted Filter

$$\frac{1}{16}\times\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

- $\sin(\theta+2\pi)=\sin(\theta)$
- $\cos(\theta+2\pi)=\cos(\theta)$

$\sin(\theta+\pi/2)=\cos(\theta)$
$\cos(\theta+\pi/2)=-\sin(\theta)$

Deg/Rad	$\sin(x)$	$\cos(x)$
0° or 0, 360° or 2π	0	1
30° or $\frac{\pi}{6}$	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$
45° or $\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$
60° or $\frac{\pi}{3}$	$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$
90° or $\frac{\pi}{2}$	1	0
120° or $\frac{2\pi}{3}$	$\frac{\sqrt{3}}{2}$	$-\frac{1}{2}$
135° or $\frac{3\pi}{4}$	$\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$
150° or $\frac{5\pi}{6}$	$\frac{1}{2}$	$-\frac{\sqrt{3}}{2}$
180° or π	0	-1

- Digital image → discrete function ของ 2 variables (x,y)
- **1D DFT**:

$$F(u)=\frac{1}{M}\sum_{x=0}^{M-1}f(x)e^{-j2\pi ux/M}$$

$$f(x)=\sum_{u=0}^{M-1}F(u)e^{j2\pi ux/M}$$

Euler's Formula
$e^{j\theta}=\cos\theta+j\sin\theta$

- **2D DFT** สำหรับภาพ MxN:

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

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

Real image → FT conjugate symmetric F(u,v) = F*(-u,-v)
Shifting → คูณด้วย (-1)^(x+y) กับ ภาพก่อนค่อย FT ได้แบบ shift center เลย
Convolution theorem: convolution ใน spatial domain ↔ multiplication ใน frequency

Sampling & Quantization : From continuous sensed data to digital form

- **Sampling** → digitizing coordinates. **position**
  - จากภาพที่มีลายละเอียดยิบย่อย แปะ grid แล้วเอาค่าเฉพาะตำแหน่งที่ตรง grid , Resolution
- **Quantization** → digitizing amplitude. **value**
  - map จากค่าที่ละเอียดนับไม่ถ้วนเป็นสีต่างๆ , Bit-depth
- Digital image: `f(x,y)` with M rows, N columns, pixel intensities `[0 … L-1]`.
- **Resolutions**:
  - Spatial resolution → smallest discernable detail.
  - Gray-level resolution → smallest intensity difference (e.g., 8-bit = 256 levels). L = 256

- ตัด high-frequency → blur, smooth

Low pass filters in Freq

- Types:
  - **Ideal LPF**: binary mask (cutoff radius D0) D0 น้อย = blur
  - **Butterworth LPF**: smooth transition, order n n ยิ่งมากยิ่งชัน
  - **Gaussian LPF**: smoothest, no ringing D0 น้อย = blur

High pass filters in Freq

- ตัด low-frequency → sharpen edges
- Ideal / Butterworth / Gaussian HPF = 1 - LPF, 3 ประเภทเหมือน LPF

- ภาพที่เสื่อมสภาพเกิดจาก
  - Noise (external disturbance)
  - Blur (out-of-focus, motion blur)
- Model (spatial domain):

$$g(x,y)=h(x,y)*f(x,y)+\eta(x,y)$$

- ใน frequency domain:

$$G(u,v)=H(u,v)F(u,v)+N(u,v)$$

3. Color Fundamentals

- Color = powerful attribute for recognition.
- **Achromatic light** → intensity.
- **Chromatic light (400–700nm)**:
  - Radiance = source energy (sun, light)
  - Luminance = observed energy (observed energy got from obj)
  - Brightness = perceived color sensation (color sensation in retina)
- **Human Vision**:
  - Cones: ~65% red, 33% green, 2% blue. (but blue most sensitive)
  - Colors form from variable RGB (Primary color) combinations.
  - Secondary Color : Magenta, Cyan, Yellow
- Attributes:
  - **Brightness (Intensity)** : Achromatic component
  - **Hue (Dominant Wavelength)** : Chromatic component
  - **Saturation (Amount of white light)** : Chromatic component
  - Hue and Saturation : **Chromaticity**

How to represent color

- **CIE Chromaticity Diagram** → standard color representation.
  - z = 1 - (x + y) , x = red , y = green
- **Tristimulus** : Amount of red, green and blue to form color
  - $x=\frac{X}{X+Y+Z}$, $y=\frac{Y}{X+Y+Z}$, $z=\frac{Z}{X+Y+Z}$
  - $x+y+z=1$

88	89	96	108
77	77	84	98
63	70	83	97
57	65	80	89

average | difference in col

88.5	102
77	91
66.5	90
61	84.5

-1	-12
0	-14
-7	-14
-8	-9

-show vertical edge

82.75	96
63.75	87.25

Horizontal

-11.5	11
-5.5	5.5

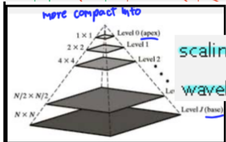
Vertical

-0.5	-13
-7.5	-10.5

Diagonal

-1	2
1	-6

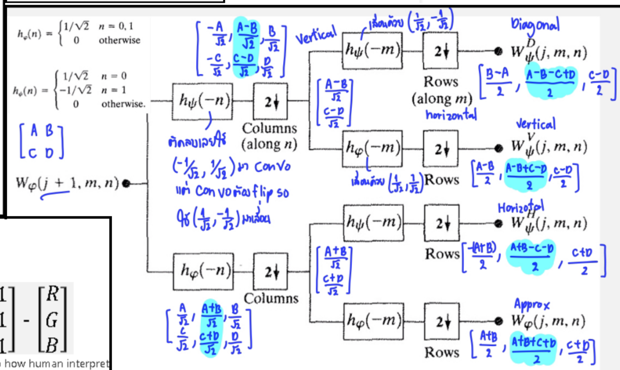
more compact info



scaling function coefficients
wavelet function coefficients

$$h_\varphi(n)=\begin{cases}1/\sqrt{2} & n=0,1\\ 0 & \text{otherwise}\end{cases}$$

$$h_\psi(n)=\begin{cases}1/\sqrt{2} & n=0\\ -1/\sqrt{2} & n=1\\ 0 & \text{otherwise.}\end{cases}$$

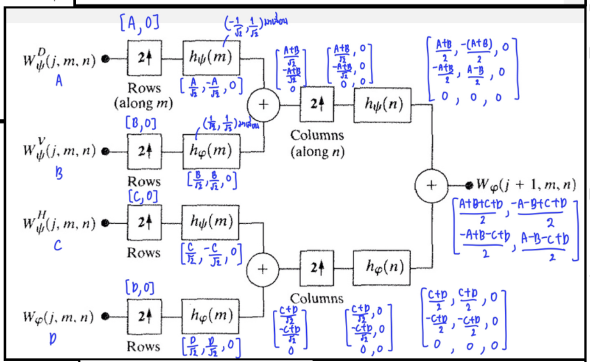


Color Models

- **RGB**: additive, displays/screens.
  - All color normalised to [0,1], 0 is black 1 is white
  - Additive = light mixing → white when maxed
- **CMY/CMYK**: subtractive, printing. Secondary Color of light
  - Subtractive = pigment/ink mixing → black when maxed

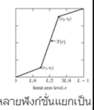
$$\begin{bmatrix}C\\M\\Y\end{bmatrix}=\begin{bmatrix}1\\1\\1\end{bmatrix}-\begin{bmatrix}R\\G\\B\end{bmatrix}$$

- **HSI/HSV**: separates hue, saturation, intensity – aligns with human perception. close to how human interpret colors, ideal tool for image processing algorithm.
  - Hue : pure color
  - Saturation : degree hue diluted by white light
  - Brightness : Intensity
- **Gray** = 0.3R + 0.6G + 0.11B
- **L*a*b***/CIELAB model: device-independent, L* = lightness, a* = green-red, b* = blue-yellow.



5. Arithmetic Operations in Spatial Domain

- Operates directly on pixels: 1. piece-wise 2.neighboring
  - `g(x,y) = T[f(x,y)] = s = T(r)`
- **Point Operators**: multiplication (gain), addition (bias).
  - Gain controls contrast. เพราะทำให้ผลต่างระหว่าง pixel ที่ต่างเพิ่มขึ้น
  - Bias controls brightness. เพราะทำให้ค่าเพิ่มขึ้นหมด
  - `g(x,y) = af(x,y)+ b`
- **Gray-Level Transformations**: Image Enhancement $s=(L-1)-r$ (255 / maximum)
  1. Negative transformation `s = (L-1) - r` 0 -> 255 , 255 -> 0
  2. Log transformation (expand dark regions) `s = clog(1+r)` , r = normalised intensity
     - log0 ไม่มีค่า so + 1
     - log ทำให้ค่าลดลง -> brighter
  3. Power-law (gamma correction) $s=cr^{Y}$
     - c, Y = positive constant
     - Y > 1 Brighter, Y < 1 Darker
     - ใช้กับ MRI Y = 0.4,0.6 more detailed becomes visible
  4. Piecewise-linear (contrast stretching) ไม่ได้ใช้ function เดียวสำหรับทุก intensity แต่มีหลายฟังก์ชันแยกเป็น range
  5. Gray-level slicing : เอาเฉพาะ intensity บางส่วนที่ interest ส่วนอื่นให้มันหลายเป็น 0 หรือคงค่าเดิมไว้ เป็น Piecewise-linear แบบหนึ่ง
  6. Bit-plane slicing : แยกตาม bit เอาเฉพาะบาง bit (MSB) -> image compression plane 0 = LSB



- **Logic/Arithmetic Ops**:
  - Subtraction → background subtraction. g(x,y) = f(x,y)foreground - h(x,y)background
  - Averaging → noise reduction (astronomy). เอาหลายๆ รูปมา avg เพื่อลด noise
  - AND/OR masking → select ROIs (Region of interest).
- **Interpolation**:
  - Nearest Neighboring : Just pick a nearest one, so blocky.
  - Linear : Averaging so smooth but blurry transition
  - Bicubic : Averaging but also fit with polynomial curve, so might have exceed.
  - Area : Nearest + Averaging

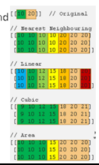


Image compression = ลดขนาดไฟล์ โดยยัง retain ข้อมูลที่จำเป็น

Compression ratio:

$C_R=\frac{ori}{compress}$ - can be any information-carrying unit

$$L_{avg}=\sum_{k=0}^{L-1}l(r_k)p_r(r_k)$$

Relative data redundancy

$R_D=1-\frac{1}{C_R}$

- Information of event E:

$$I(E)=-\log P(E)$$

- Entropy/uncertainty of source:

$$H(z)=-\sum_{j=1}^{J}P(a_j)\log_2 P(a_j)$$

- log2 ได้ผลลัพธ์หน่วยเป็น bit
- Entropy ≈ minimum average bits/pixel needed

Image f มี gray levels = 0,…,L-1
Histogram: $h(r_k)=n_k$ (จำนวน pixel ที่ intensity = r_k)
Normalized Histogram: $p(r_k)=\frac{n_k}{n}$
- เป็น estimate ของ probability
- $\Sigma p(r_k)=1$

- Idea: ทำ histogram ให้ออกมา **flat** → กระจายค่า intensity ให้ contrast ดีขึ้น
- ใช้ **PDF + CDF** ในการหา mapping function `s = T(r)`
- Formula: $s_k=(L-1)\sum_{j=0}^{k}p(r_j)$
- ทำให้ dark area → brighter, bright area → darker balance

Salt & Pepper (Impulse/shot/binary noise)
- During image digitizing -> impulse corruption
- จุดดำ-ขาวสุ่ม เหมือนเกลือพริกไทย
- PDF:

$$p(z)=\begin{cases}P_a & z=a\\ P_b & z=b\\ 0 & \text{otherwise}\end{cases}$$

Rayleigh noise
- Medical imaging -> X-ray, MRI

Exponential & Gamma
- Laser imaging

Exponential
- PET/SPECT

Uniform
- Dust in camera lens

Gaussian Noise
- เกิดจาก random fluctuation ของสัญญาณ
- พบได้บ่อยสุด เกิดจาก electronic circuit noise, sensor noise from poor illumination
- noise ก็เป็นค่า mean จึงคล้ายกับค่าในรูป ไม่ได้แหวก
- PDF:

$$p(z)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-(z-\mu)^2/(2\sigma^2)}$$

Periodic Noise
- จาก electrical / EM interference
- เห็นเป็น pattern ซ้ำๆ ใน spectrum

Speckle Noise (multiplicative)
$g(x,y)=f(x,y)\times s(x,y)$ พบใน radar, ultrasound imaging (wave interference)

(a) Spatial Filtering
- **Averaging / Low-pass filters** → ลด Salt & Pepper, Gaussian noise
  - ปัญหาเมื่อใช้ S&P คือใช้ mask ใหญ่ดีกว่าแต่ก็เบลอมากกว่า
- **Median filter** → ดีมากกับ Salt & Pepper
- **Gaussian filter** → blur เพื่อลด noise

(b) Frequency Domain Filtering
- Band-reject/pass filters → ลบเป็น freq ทั้งวง ลบ periodic noise
- Notch filters → ลบ frequency เฉพาะจุด (ต้องทำเป็นคู่ symmetric)

(c) Degradation function

Estimating Degradation function
- Image observation
- Experimentation
- Modelling

Inverse filtering
- Idea: หารด้วย H(u,v) เพื่อย้อน degradation

$$\hat{F}(u,v)=\frac{G(u,v)}{H(u,v)}$$

- ปัญหา: ถ้ามี noise → amplify noise อาจมี noise dominant

(d) Wiener Filtering
- Minimize Mean Square Error (MMSE):

$$\hat{F}(u,v)=\frac{H^*(u,v)}{|H(u,v)|^2+K}G(u,v)$$

- K = noise-to-signal ratio constant
- ใช้ได้ผลดีกว่า inverse filter

Evaluation
- PSNR(Peak signal-to-noise ratio) : ยิ่งสูงยิ่งดี
  - $PSNR=10\log_{10}\frac{(L-1)^2}{MSE}=20\log_{10}\frac{(L-1)}{RMSE}$
  - $MSE=\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(O(i,j)-D(i,j))^2$

1. ค่าของ Entropy เปลี่ยนได้หรือไม่
- ขึ้นกับ **Distribution** ของความน่าจะเป็น $P(a_j)$
- ถ้า distribution กระจายเท่ากัน (**uniform**) → entropy สูงสุด
- ถ้า distribution เบ้ไปทางใดทางหนึ่ง (บางสัญลักษณ์มีความน่าจะเป็นสูงกว่ามาก) → entropy ต่ำลง

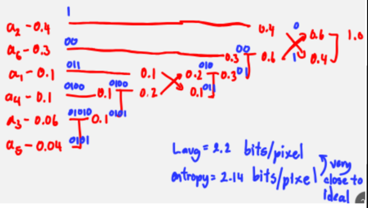
ดังนั้น entropy ไม่คงที่ มันลดได้เมื่อความไม่แน่นอนของข้อมูลลดลง

Error-Free (Lossless) Compression

code สั้นสุดให้ gray scale ที่ปรากฏบ่อยสุด ทำให้ Lavg ลดลง

1. **Variable-Length Coding**
   - Huffman coding: สร้าง tree จาก histogram probability
     - น้อยสุดสองอันรวมกันไปเรื่อยๆ จนเหลือแค่ 2
     - มากกว่าเพิ่ม 0 น้อยกว่าเติม 1

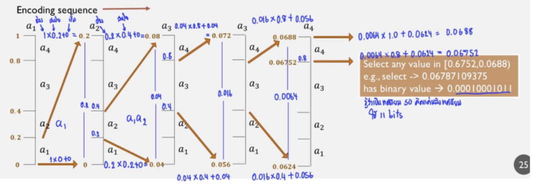
Symbol	Prob
a_1	0.1
a_2	0.4
a_3	0.06
a_4	0.1
a_5	0.04
a_6	0.3



Lavg = 2.2 bits/pixel
entropy = 2.14 bits/pixel } very close to ideal

- Arithmetic coding: encode เป็นเลข real [0,1), ใช้ distribution

• A = {a_1, a_2, a_3, a_3, a_4}; P(a_1)=0.2, P(a_2)=0.2, P(a_3) = 0.4, P(a_4) = 0.2;



2. **LZW Coding**
   - ใช้ dictionary-based compression, ใช้ 9 bits 256 เป็นต้นไปใช้เก็บ pattern ใหม่ใน dict

• Consider the 4x4, 8-bit image of a vertical edge:

```
39  39  126  126
39  39  126  126
39  39  126  126
39  39  126  126
```

processing pixels in left-to-right and top-to-bottom manner

• A 512-word dictionary with the following starting content is assumed:

Dictionary Location	Entry
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

for this picture

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

LZW coding example

3. **Bit-plane Coding**
   - แยก plane ของ bits MSB = เห็นภาพชัด, ใช้ run-length coding

4. **Run-Length Coding (RLC)**
   - ดีมากกับ binary images, encode (value, length)
   - รวมเป็นก้อนๆ สำหรับ pixel ที่มีค่าเท่ากัน
   - binary horizontal : มีแค่จำนวนก็พอ เริ่มด้วย 0
   - extended : (color, #run) : ใช้เป็น tuple pair

the run.

$8 = 2^n = 3$ bits

ต้อง 0 มาก่อนก็ได้, run: 0 ต่อ, เป็นไปไม่ได้ number of zero

0	0	0	0	0	0	0	0	8
1	1	1	1	0	0	0	0	0,4,4
0	1	1	0	0	0	0	0	1,2,5
0	1	1	1	1	1	0	0	1,5,2
0	1	1	1	0	0	1	0	1,3,2,1,1
0	0	1	0	0	1	1	0	2,1,2,2,1
1	1	1	1	0	1	0	0	0,4,1,1,2
0	0	0	0	0	0	0	0	8

3 (1+3+3+3+5+5+5+1) = number of bit needed

8×8 4-bit image

4bit 3bit

10	10	10	10	10	10	10	10	(10,8)
10	10	10	10	10	12	12	12	(10,5)(12,3)
10	10	10	10	10	12	12	12	same
0	0	0	10	10	10	0	0	(0,3)(10,3)(0,2)
5	5	5	0	0	0	1	0	(5,3)(0,3)(1,1)(0,1)
5	5	5	10	10	9	9	10	(5,3)(10,2)(9,2)(10,1)
5	5	5	4	4	0	0	0	(5,3)(4,3)(2,2)
0	0	0	0	0	0	0	0	(0,8)

5. **Lossless Predictive Coding**
   - ใช้ค่าคาดเดาจากเพื่อนบ้าน → encode error

Lossy Compression

1. **Lossy Predictive Coding**
   - เพิ่ม quantizer → บีบ error เข้าช่วงจำกัด
2. **Transform Coding**
   - ใช้ DCT (Discrete Cosine Transform) → JPEG baseline
     - เนื่องจากใช้เฉพาะ real number เลยเป็น lossy
   - ใช้ Wavelet → JPEG2000

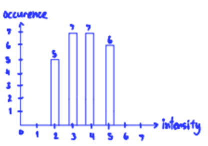
1) Image Histogram (6 points) L=8

1.1) A 5x5 digital image, $f(x,y)$ for intensity in the range [0, 7] is shown below. Calculate and draw a histogram of this image with bin size = 1

2	3	4	5	5
2	3	4	5	5
2	3	4	4	4
2	3	5	3	3
2	3	5	4	4

$f(x,y)$

h(2) = 5
h(3) = 7
h(4) = 7
h(5) = 6

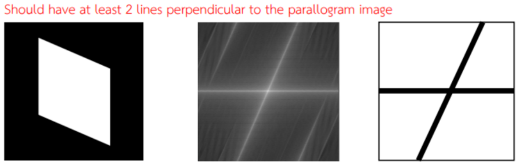


1.2) Analyze the characteristics of the image in terms of contrast and brightness

Ans: Histogram is closely distributed between 2-7 so this image has a low contrast with a grayish color tone.

Ans. DFT, students can draw just the rightmost pattern

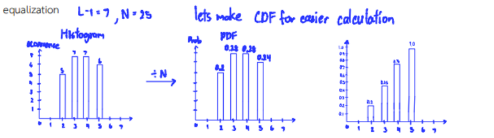
Should have at least 2 lines perpendicular to the parallogram image

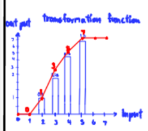


Wavelet transform

Show how to perform histogram equalization, draw a transformation function, and draw a new image after equalization L-1=7, N=25

lets make CDF for easier calculation





new image

1	3	5	7	7
1	3	5	7	7
1	3	5	5	5
1	3	7	3	3
1	3	7	5	5

To nearest integer

0 = 0 → 0
1 = 0 → 0
2 = 0.2×7 → 1
3 = 0.45×7 → 3
4 = 0.76×7 → 5
5 = 1×7 → 7
6 = 1×7 → 7
7 = 1×7 → 7

$$\left(\frac{n_0+n_1+\cdots+n_j}{N}\right)(L-1)$$

2.1) Given a small 3x3 image, $f(x,y)$, show how to sharpen an image using 3x3 Laplacian filter (with either positive or negative center) and image boundary pixels are patched with zero. (In this case, intensity can be negative value; positive value can be more than 255)

3	3	3
3	2	2
3	2	1

$f(x,y)$

Laplacian filter :
$$\begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix}$$

add with original got sharpening :
$$\begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{matrix}$$

0	0	0	0	0
0	3	3	3	0
0	3	2	2	0
0	3	2	1	0
0	0	0	0	0

0,0 = 0+0+15-3-3 = 9
0,1 = 0-3+15-3-2 = 7
0,2 = 0-3+15+0-2 = 10
1,0 = -3+0+15-3-3 = 7
1,1 = -3-3+10-2-2 = 0
1,2 = -3-2+10+0-1 = 4
2,0 = -3+0+15-2+0 = 10
2,1 = -2-3+10-1+0 = 4
2,2 = -2-2+5+0+0 = 1

The enhanced image

9	7	10
7	0	4
10	4	1

Enhanced image show higher diff of neighboring pixel

2.2) Describe the difference between the enhanced image and the original image.

Ans. The enhanced image show higher differences in intensity levels of the neighboring pixels. The maximum difference of neighboring pixel intensity of the original image is 1 intensity level; whereas the max. differences in the enhanced image pixels are in the range of 3 to 7 intensity levels.

3) (3 points) A 3x3 image, $g(x,y)$ is the 4-bit image which is corrupted by salt and pepper noise.

15	15	2	3	3
15	15	2	3	3
1	1	2	0	0
1	1	2	3	3
1	1	2	3	3

$g(x,y)$

3.1) Show how to restore the image from the noise by using 3x3 median filtering with replicate border padding technique.

median = position 5

1 1 2 2 2 15 15 15 15
0 1 1 2 2 3 3 15 15
0 0 2 2 2 3 3 3 3
1 1 1 1 2 2 2 15 15
0 1 1 2 2 2 3 3 15
0 0 2 2 2 3 3 3 3
1 1 1 1 1 1 2 2 2
0 1 1 1 2 2 2 3 3
0 0 2 1 2 3 3 3 3

The restored image ($\hat{f}(x,y)$)

2	2	2
2	2	2
1	2	2

3.2) Assuming you know original image $f(x,y)$ below, compare PSNR of the noisy image and the restored image. What is the purpose of PSNR? **original > restore > noisy เพราะ MSE เพิ่มค่าลด**

PSNR of noisy image should be less than the restored image because the mean squared error or the divider in the formula is larger; the total PSNR will be smaller. In contrast for the more similar images of the restored image and the original image, PSNR will be higher due to smaller mean squared error as the divider.

4. (6 points) Given an 8-bit 3x3 image, $f(x,y)$ below, $F(u,v)$ is the Fourier transform of $f(x,y)$.

4.1) Show how to calculate the Fourier transform at 1 and 2 in the image below.

9	0	0
0	18	0
0	0	9

$f(x,y)$

	0,0	0,+1	0,-1
	1 a	-0.5-0.87 j b	-0.5+0.87 j c
	-0.5-0.87 j d	-0.5+0.87 j e	4 f
	2 g	4 h	-0.5-0.87 j i

$F(u,v)$

-1,-1 i	-1,0 g	-1,+1 h
0,-1 c	a	b 0,+1
f +1,-1	d +1,0	e +1,+1

① $\frac{1}{9}\sum_{x=0}^{2}\sum_{y=0}^{2} f(x,y)\, e^{-j2\pi(\frac{ux}{3}+\frac{vy}{3})} = \frac{36}{9} = 4$

4.2) What is the Fourier Spectrum at 1 and 2?

② $\frac{1}{9}\sum_{x=0}^{2}\sum_{y=0}^{2} f(x,y)\, e^{-j2\pi(\frac{2x}{3})}$ u=2, v=0

x,y
0,0 9×1 = 9
1,1 $18 \times e^{-j\frac{4\pi}{3}} = 18(\cos\frac{-4\pi}{3} + j\sin\frac{-4\pi}{3}) = 18(\cos\frac{4\pi}{3} - j\sin\frac{4\pi}{3}) = 18(-\frac{1}{2}+\frac{\sqrt{3}}{2}j)$
2,2 $9 \times e^{-j\frac{8\pi}{3}} = 9(\cos\frac{8\pi}{3} - j\sin\frac{8\pi}{3}) = 9(-\frac{1}{2}-\frac{\sqrt{3}}{2}j)$

sin -θ = -sin θ
cos -θ = cos θ

$= 9 - 13.5 + 4.5\sqrt{3}j = \frac{-4.5+4.5\sqrt{3}j}{9} = -0.5+0.5\sqrt{3}j$

4.3) Determine the shifted Fourier transform

g h i → i g h
a b c → c a b
d e f → f d e

-0.5-0.87j	-0.5+0.87j	4
-0.5+0.87j	4	-0.5-0.87j
4	-0.5-0.87j	-0.5+0.87j

4.4) Design a 3x3 ideal high pass filter with cutoff frequency (D_0) = 1 pixel. Due to the small size of the filter, use the condition below.

$$H(u,v) = \begin{cases} 0 & if\ D(u,v) < D_0 \\ 1 & if\ D(u,v) \geq D_0 \end{cases}$$

1 1 = $\sqrt{2}$
0 1 = 1
0 0 = 0

where $H(u,v)$ is the ideal high-pass filter, $D(u,v)$ is distance of (u,v) from image center.

Ideal high-pass Filter

1	1	1
1	0	1
1	1	1

4.5) From Fourier transform of an image in 4.1), show how to compute the result of high-pass filtering in frequency domain. Just multiply it, only center became zero

4.6) After passing the high pass filter, then calculate the inverse Fourier Transform of $F(u,v)$ to get the filtered image in spatial domain, $f(x,y)$. Analyze the characteristics of $f(x,y)$. (Do not need to calculate iDFT of $F(u,v)$) highlight all edge

No DC and highlight all edge cuz it is high-pass filter

7.5) Determine the average number of bits (L_{avg}) required to code Code #1 and Code #2 using

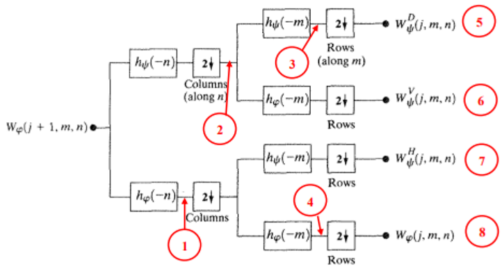
$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) P_k(r_k)$$

where $l(r_k)$ is bit-length of intensity level, r_k and $P_k(r_k)$ is the probability of r_k occurs in the image.

Code #1 $L_{avg} = 2$

Code #2 $L_{avg} = \frac{4}{28}\times 3 + \frac{8}{28}\times 2 + \frac{3}{28}\times 3 + \frac{13}{28}\times 1$

$= \frac{12}{28} + \frac{16}{28} + \frac{9}{28} + \frac{13}{28} = \frac{22}{28} = 1.785$ #



Analysis filter bank

6.1) How many and what are the levels of image pyramid are used in this case?

2 level, cuz j=0 it only got one pixel, and j=1

3	-1
1	-3

A 2x2 image, $f(x,y)$

6.2
① $* [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] = \begin{bmatrix} \frac{3}{\sqrt{2}}, \frac{2}{\sqrt{2}}, \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}}, \frac{-2}{\sqrt{2}}, \frac{-3}{\sqrt{2}} \end{bmatrix}$ (1)

② $* [-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] = -\frac{3}{\sqrt{2}}, \frac{4}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \quad -\frac{1}{\sqrt{2}}, \frac{4}{\sqrt{2}}, -\frac{3}{\sqrt{2}} = \begin{bmatrix} \frac{4}{\sqrt{2}} \\ \frac{4}{\sqrt{2}} \end{bmatrix}$ down sampling convo ต้อง flip

③ $* [-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] = -\frac{4}{2}, 0, \frac{4}{2} = [-2, \boxed{0}, 2]$ (5)

④ $* [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] = \frac{2}{2}, 0, \frac{-2}{2} = [1, \boxed{0}, -1]$ (8)

⑥ $* [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] = \frac{4}{2}, \frac{8}{2}, \frac{4}{2} = 2, \boxed{4}, 2$ (6)

⑦ $* [-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] = -\frac{2}{2}, \frac{4}{2}, \frac{-2}{2} = -1, \boxed{2}, -1$ (7)

6.3) Any zero in wavelet coefficients? If so, explain why it is zero?

$\begin{bmatrix} A & H \\ V & D \end{bmatrix}$ A : 0 cuz avg = 0 > No DC component
D : different in diagnal is cancelling

7) A gray-level 8-bit image of 4 x 7 pixels is shown below

0	100	200	250	250	250	100
50	100	150	250	250	250	100
50	100	150	250	250	250	100
50	100	150	250	250	250	100

7.1) The image intensity levels are equally quantized from 8-bit to have 4 intensity levels (0, 1, 2 and 3). Write the quantized image below.

0	1	3	3	3	3	1
0	1	2	3	3	3	1
0	1	2	3	3	3	1
0	1	2	3	3	3	1

8bit = 0-255
$2^{8-2} = 2^6 = 64$
0-63 = 0
64-127 = 1
128-191 = 2
192-255 = 3

7.2) What is the entropy of the quantized image compared with the original image?

↳ $H(z) = -\sum p(z) \log_2 p(z)$

Quantized
p(0) = 4/28 = 1/7
p(1) = 8/28 = 2/7
p(2) = 3/28
p(3) = 13/28

H_Q = 0.401 + 0.516 + 0.345 + 0.514 = 1.776

Original
p(0) = 1/28
p(50) = 3/28
100 - 8/28 = 2/7
150 = 3/28
200 = 1/28
250 = 12/28

H_o = 0.172 + 0.345 + 0.516 + 0.345 + 0.172 + 0.524 = 2.074

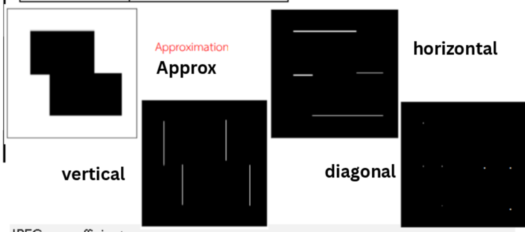
7.3) Represent these 4 levels with minimum fixed bits encoded in "**Code #1**" in the table below. And find the probability of each intensity level r_k at kth level.

Intensity level (r_k)	Code #1	Probability of r_k occurs $P(r_k)$
0	00	1/7
1	01	2/7
2	10	3/28
3	11	13/28

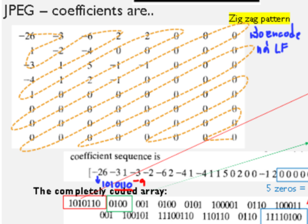
7.4) Implement Huffman Coding of the 4 intensity levels in 7.1) and encode in terms of "**Code #2**" in the table below.

Intensity level (r_k)	Code #2
0	010
1	00
2	011
3	1

p(3) = 13/28 1
p(1) = 8/28 00
p(0) = 4/28 010
p(2) = 3/28 011
7/28 01
15/28 0



JPEG – coefficients are..



DC values of the previous block is -17
-26-(-17) = -9 (Cat 4 in Table 2)
Base code is 101 ____ (Table 3)
9 -> 1001
-9 -> 0110 (1' complement of 9)
Code = 1010110

AC -3 is in Cat 2 in Table 2, run/cat = 0/2
Base code is 01
3 -> 11, -3 -> 00 (1' complement of 3)
Code = 0100

Run/cat = 5/1
Base code = 1111010
-1 -> 0
Code = 11110100