

VII. Data Compression (A)

◆壓縮的通則：

利用資料的一致性

資料越一致的資料，越能夠進行壓縮

[References]

- 酒井善則，吉田俊之原著，原島博監修，白執善編譯，“影像壓縮術”，全華印行，2004.
- 戴顯權，“資料壓縮 Data Compression,” 旗標出版社，2007.
- I. Bocharova, *Compression for Multimedia*, Cambridge, UK, Cambridge University Press, 2010.
- D. Salomon, *Introduction to Data Compression*, Springer, 3rd ed., New York , 2004.

◎ 7-A 壓縮的哲學：

(1) 利用資料的一致性，規則性，與可預測性

(exploit redundancies and predictability, find the compact or sparse representation)

(2) 通常而言，若可以用比較精簡的自然語言來描述一個東西，那麼也就越能夠對這個東西作壓縮

Q: 最古老的壓縮技術是什麼？

(3) 資料越一致，代表統計特性越集中

包括 Fourier transform domain, histogram, eigenvalue 等方面的集中度

Data type	Compression technique	Compression rate
Audio	*.mp3	
Image	*.jpg	
Video	*.mpg *.mpeg *.mp4 *.avi *.wmv *.mov	

思考：如何對以下的資料作壓縮

Article:

Song:

Cartoon or Mark:

Compression: Original signal → Compact representation + residual information

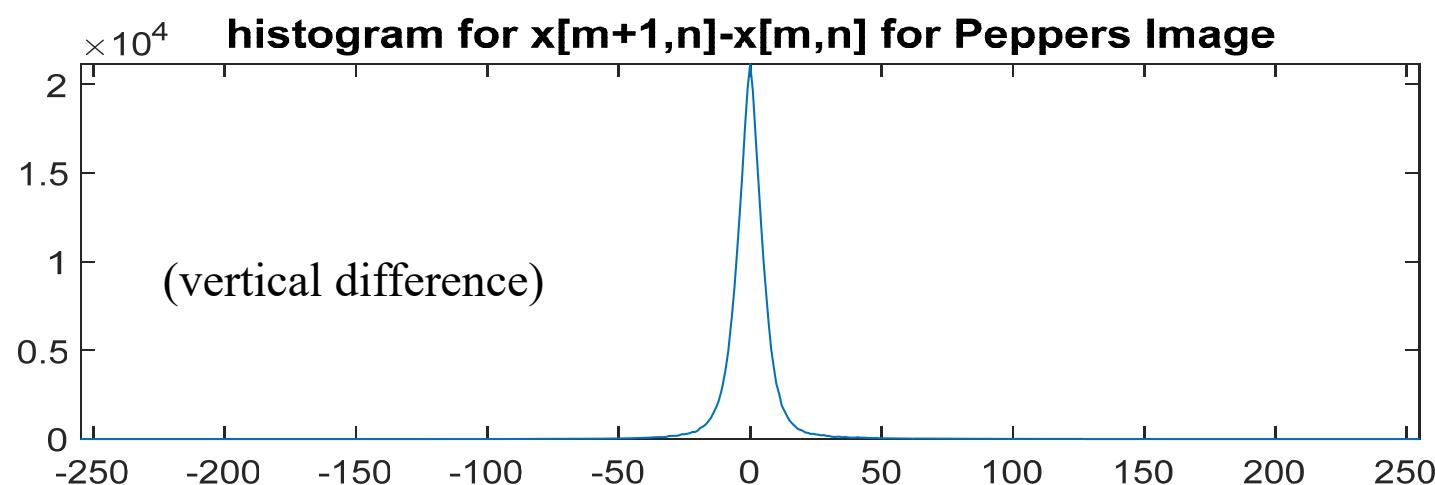
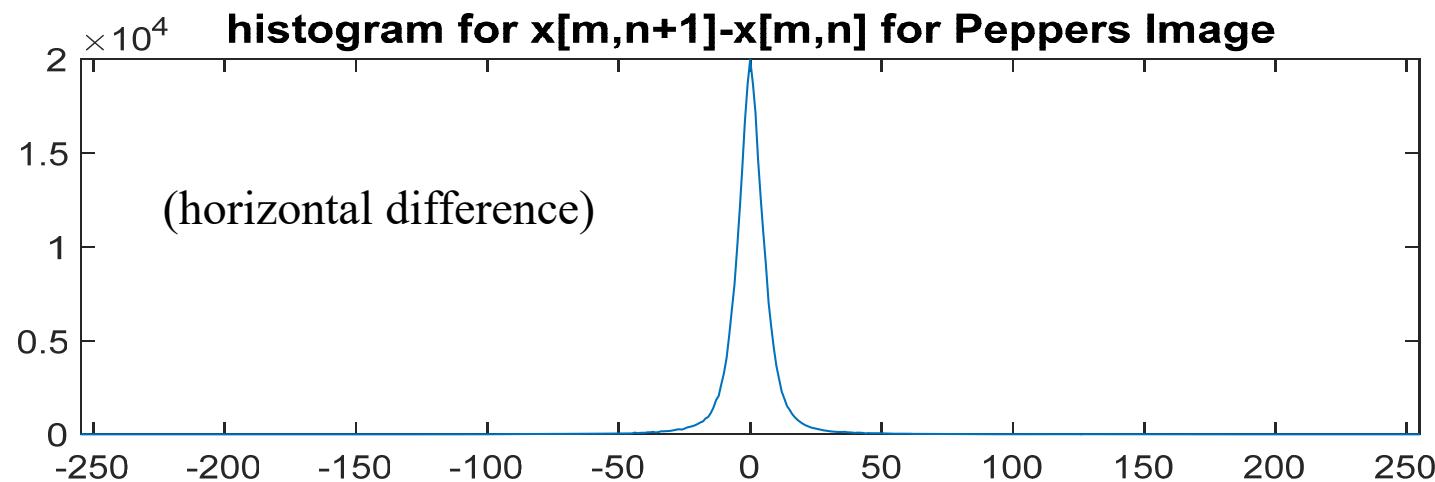
◎ 7-B Compression for Images

- 影像的「一致性」：

Space domain: 每一點的值，會和相鄰的點的值非常接近

$$F[m, n+1] \approx F[m, n], \quad F[m+1, n] \approx F[m, n]$$

Frequency domain: 大多集中在低頻的地方。



Histogram:

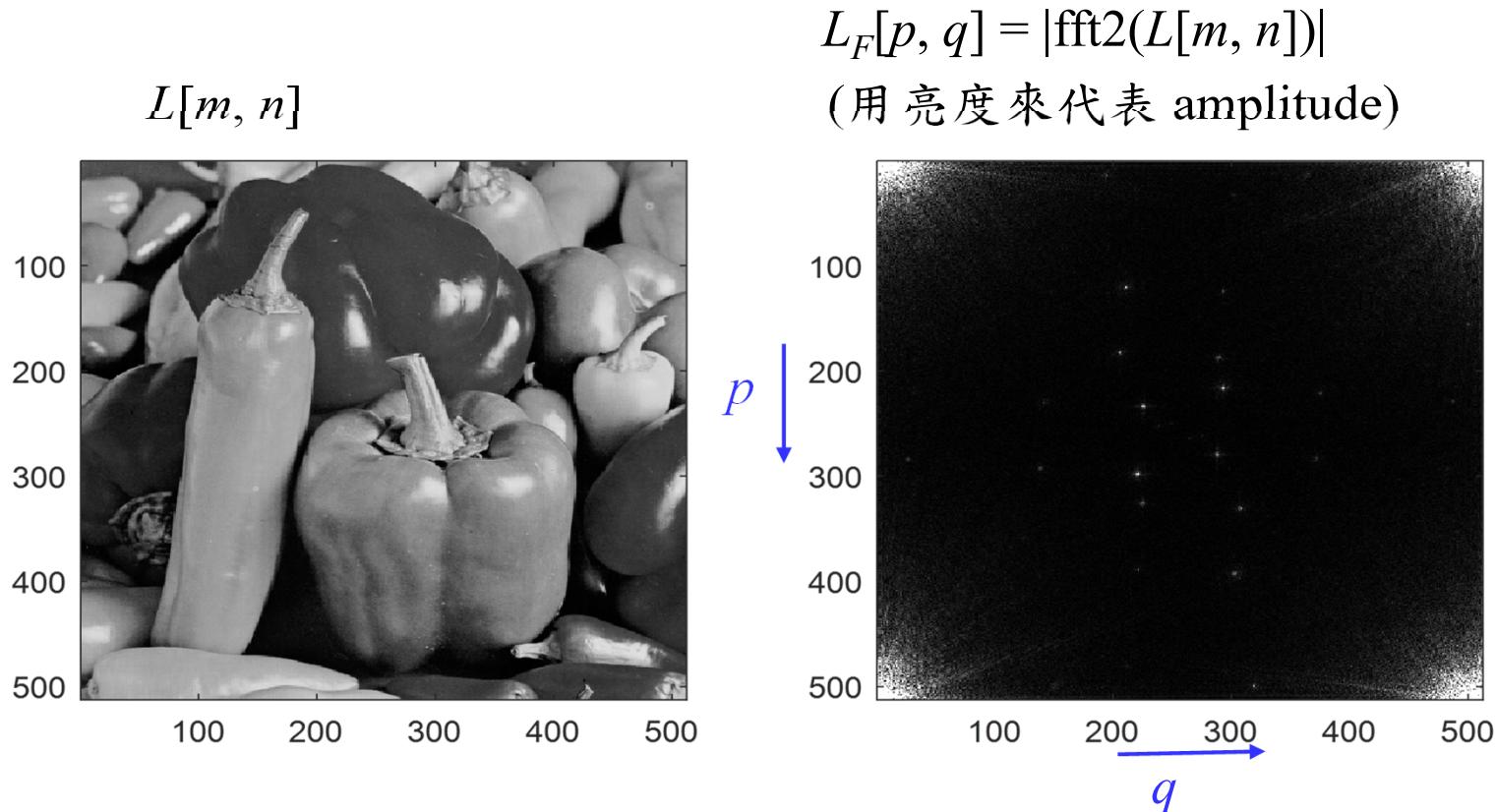
一個 vector 或一個 matrix 當中，有多少點會等於某一個值

例如： $x[n] = [1\ 2\ 3\ 4\ 4\ 5\ 5\ 3\ 5\ 5\ 4]$

則 $x[n]$ 的 histogram 為

$$h[1] = 1, h[2] = 1, h[3] = 2, \quad h[4] = 3, h[5] = 4$$

Lena Image 頻譜 (frequency domain) 的一致性



$$L_F[p, q] = \text{fft2}\{L[m, n]\} = \sum_{m=1}^M \sum_{n=1}^N L[m, n] e^{-j2\pi \frac{pm}{M}} e^{-j2\pi \frac{qn}{N}}$$

$$L_F[p, q] = L_F[p + M, q] = L_F[p, q + N] = L_F[p + M, q + N]$$

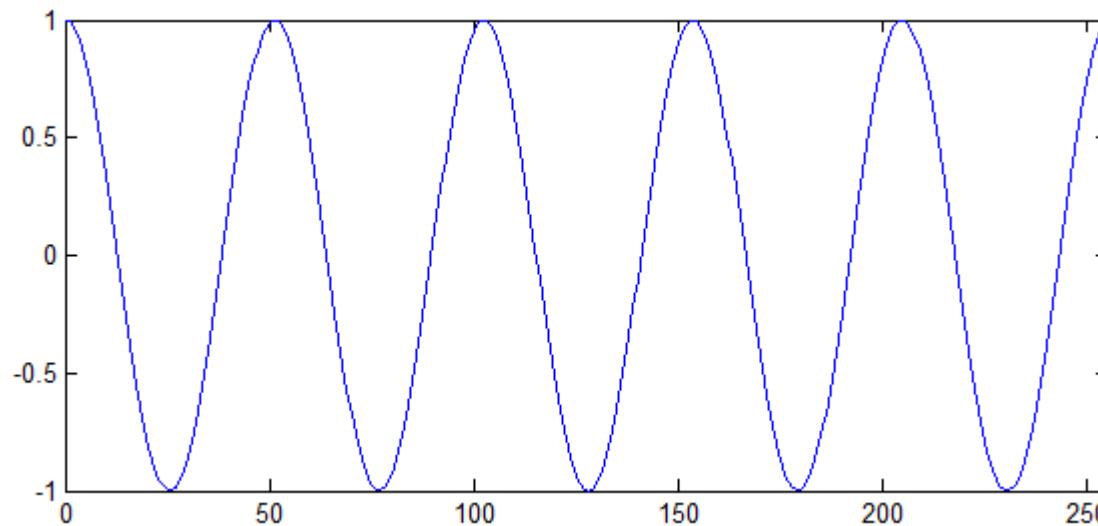
影像的「頻率」：frequency in the space domain

$$e^{j2\pi \frac{pm}{M}}$$

從 $m = 0$ 至 $m = M-1$ 之間有 p 個週期

$$p = 5$$

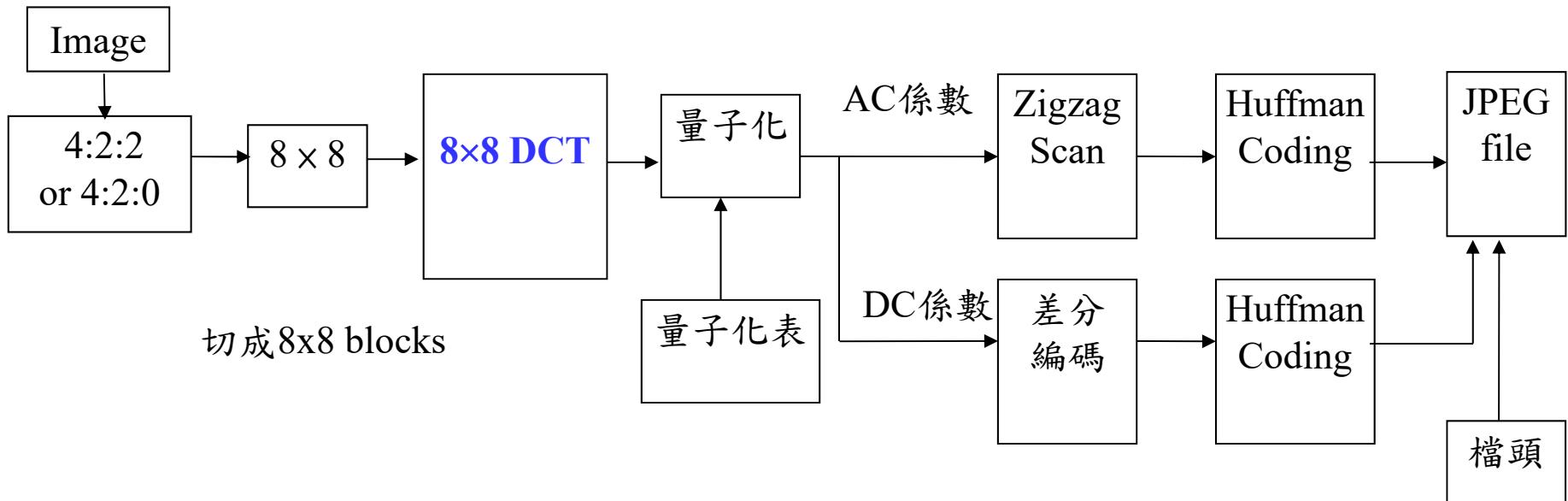
$$\text{Re}\{e^{j2\pi \frac{pm}{M}}\}$$



larger p : more variation in the space domain

◎ 7.C JPEG Standard

Process of JPEG Image Compression



- 主要用到四個技術：
 - (1) 4:2:2 or 4:2:0 (和 space domain 的一致性相關)
 - (2) 8×8 DCT (和 frequency domain 的一致性相關)
 - (3) 差分編碼 (和 space domain 的一致性相關)
 - (4) Huffman coding (和 lossless 編碼技術相關)

JPEG：影像編碼的國際標準 全名：Joint Photographic Experts Group

JPEG 官方網站：<http://www.jpeg.org/>

參考論文：G. K. Wallace, “The JPEG still picture compression standard,” *IEEE Transactions on Consumer Electronics*, vol. 38, issue 1, pp. 18-34, 1992.

JPEG 的 FAQ 網站：<http://www.faqs.org/faqs/jpeg-faq/>

JPEG 的免費 C 語言程式碼：

<http://opensource.apple.com/source/WebCore/WebCore-1C25/platform/image-decoders/jpeg/>

一般的彩色影像，可以壓縮 20 倍。

簡單的影像甚至可以壓縮超過 30 倍。

- 壓縮的技術分成兩種

lossy compression techniques

無法完全重建原來的資料

Examples: DFT, **DCT**, KLT (with quantization and truncation),
4:2:2 or 4:2:0, polynomial approximation

壓縮率較高

lossless compression techniques

可以完全重建原來的資料

Examples: binary coding, Huffman coding, arithmetic coding,
Golomb coding

壓縮率較低

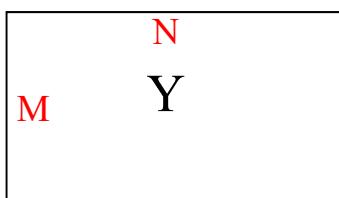
● 7-D 4:2:2 and 4:2:0

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

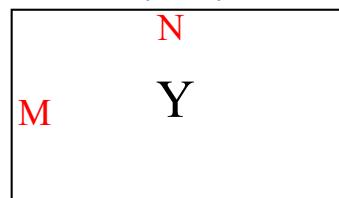
R : red, G : green, B : blue

Y : 亮度, C_b : $0.565(B-Y)$, C_r : $0.713(R-Y)$,

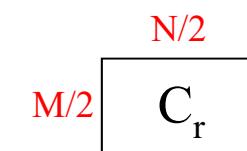
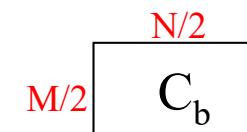
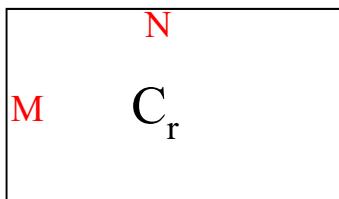
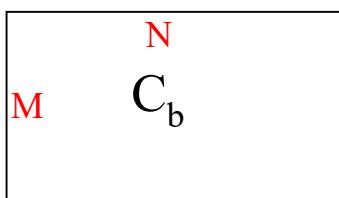
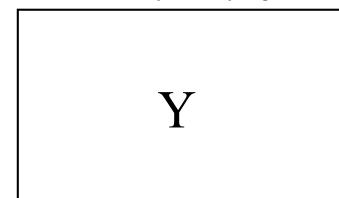
4 : 4 : 4



4 : 2 : 2



4 : 2 : 0

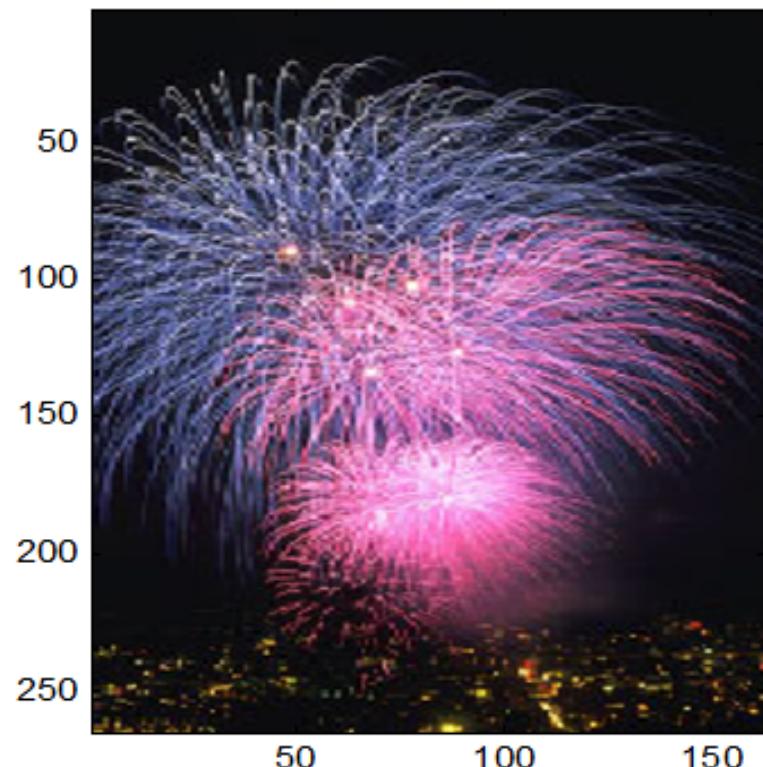


24 bits/pixel → 16 bits/pixel → 12 bits/pixel

同樣使資料量省一半的(b)(d)圖，(d)圖和原來差不多，
然而(b)圖邊緣會有失真現象。

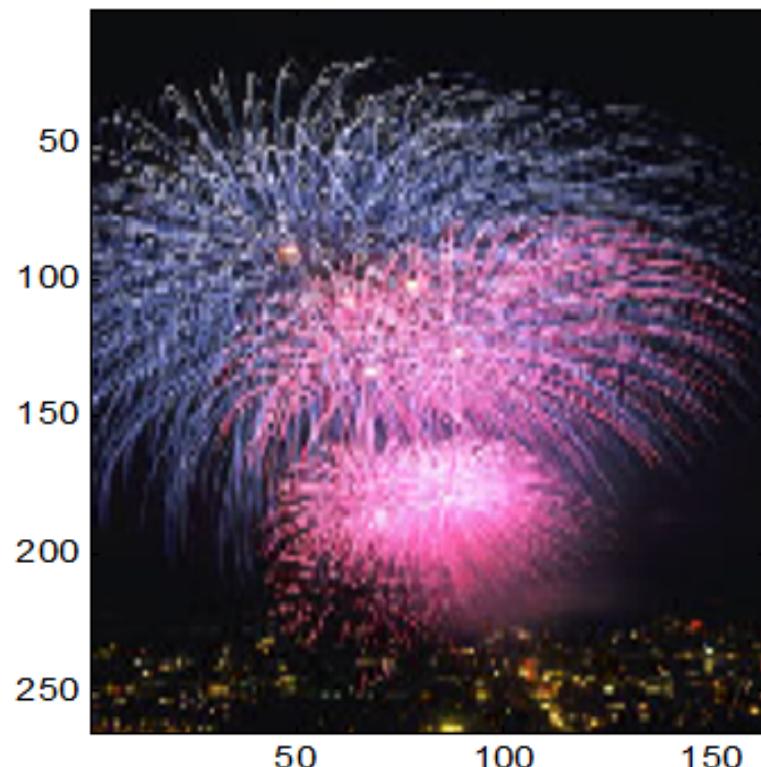
還原時，用 interpolation 的方式

原圖



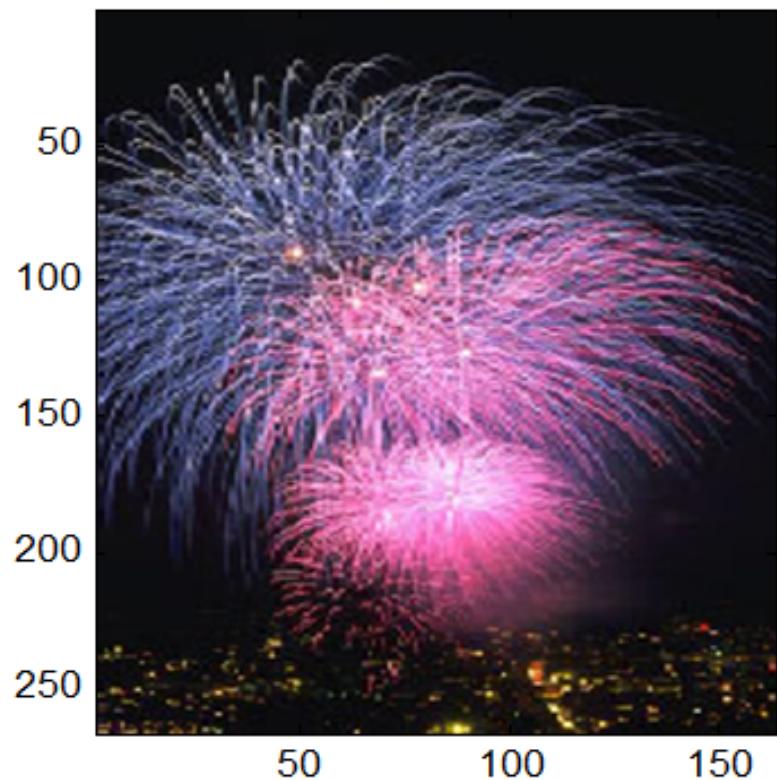
(a)

直接在縱軸取一半的pixels 再還原



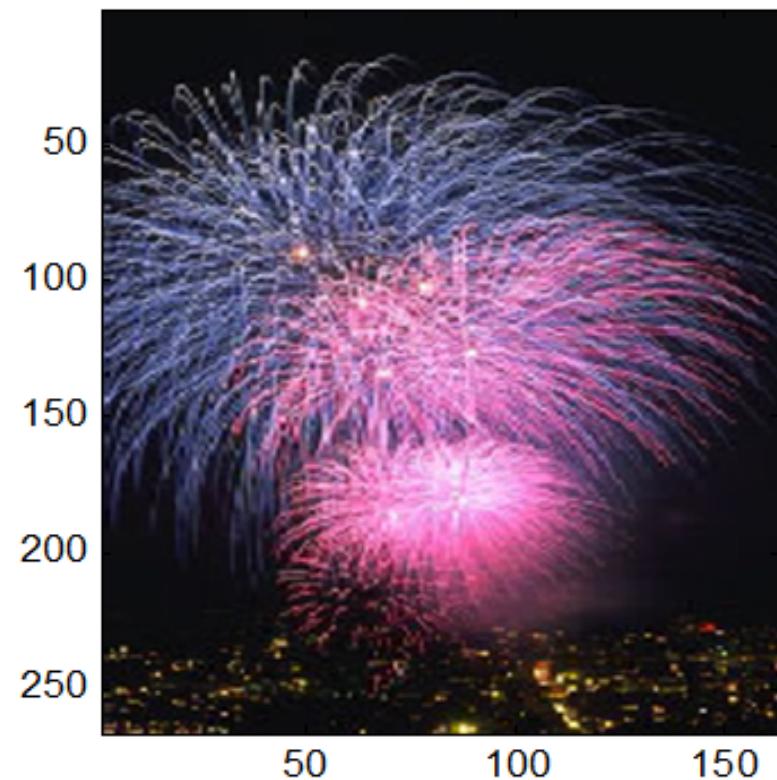
(b)

4 : 2: 2



(c)

4 : 2: 0



(d)

◎ 7-E Optimal Transform--KLT

複習：DFT 的優缺點

- Karhunen-Loeve Transform (KLT)
(similar to Principal component analysis (PCA))

It is optimal, but dependent
on the input

經過轉換後，能夠將影像的能量分佈變得最為集中

分析影像的主要成份，第二主要成份，第三主要成份，.....

- 1-D Case $X[u] = \sum_{n=0}^{N-1} x[n] K[u, n]$

$$K[u, n] = e_n[u] \quad (K = [e_0, e_1, e_2, \dots, e_{N-1}]^T)$$

e_n 為 covariance matrix \mathbf{C} 的 eigenvector

$$C[m, n] = \text{cov}(x[m], x[n]) = E[(x[m] - \bar{x}[m])(x[n] - \bar{x}[n])]$$

mean

Note: cov 代表 covariance

KLT 的理論基礎：

經過 KLT 之後，當 $u_1 \neq u_2$ 時， $X[u_1]$ 和 $X[u_2]$ 之間的 covariance 必需近於零 (即 decorrelation)

$$\text{即 } \text{cov}(X[u_1], X[u_2]) = E[(X[u_1] - \overline{X[u_1]})(X[u_2] - \overline{X[u_2]})] = 0$$

If we set

$$\mathbf{x} = [x[0], x[1], x[2], \dots, x[N-1]]^T \quad \mathbf{X} = [X[0], X[1], X[2], \dots, X[N-1]]^T$$

and

$$\mathbf{C} = E((\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T) \quad \text{where} \quad \bar{\mathbf{x}} = E(\mathbf{x})$$

$$\mathbf{C}_x = E((\mathbf{X} - \bar{\mathbf{X}})(\mathbf{X} - \bar{\mathbf{X}})^T) \quad \text{where} \quad \bar{\mathbf{X}} = E(\mathbf{X})$$

then

$$\mathbf{C}[m, n] = \text{corr}(x(m), x(n)) \quad \mathbf{C}_x[u_1, u_2] = \text{corr}(X(u_1), X(u_2))$$

\mathbf{C}_x should be a diagonal matrix.

Also note that

$$\mathbf{X} = \mathbf{K}\mathbf{x} \quad \bar{\mathbf{X}} = \mathbf{K}\bar{\mathbf{x}}$$

$$\mathbf{C}_x = E\left((\mathbf{X} - \bar{\mathbf{X}})(\mathbf{X} - \bar{\mathbf{X}})^T\right)$$

Since $\mathbf{X} = \mathbf{K}\mathbf{x}$ $\bar{\mathbf{X}} = \mathbf{K}\bar{\mathbf{x}}$

$$\begin{aligned}\mathbf{C}_x &= E\left((\mathbf{K}\mathbf{x} - \mathbf{K}\bar{\mathbf{x}})(\mathbf{K}\mathbf{x} - \mathbf{K}\bar{\mathbf{x}})^T\right) = E\left(\mathbf{K}(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{K}^T\right) \\ &= \mathbf{K}E\left((\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T\right)\mathbf{K}^T = \mathbf{K}\mathbf{C}\mathbf{K}^T\end{aligned}$$

To make \mathbf{C}_x a diagonal matrix, the KLT transform matrix \mathbf{K} should diagonalize \mathbf{C} . Suppose that the rows of \mathbf{K} are the eigenvectors of \mathbf{C} is:

$$\mathbf{C} = \mathbf{E}\mathbf{D}\mathbf{E}^T$$

where each column of \mathbf{E} is an orthonormalized eigenvector of \mathbf{C} and each diagonal entry of the diagonal matrix \mathbf{D} is the eigenvalue, then we can set

$$\mathbf{K} = \mathbf{E}^T$$

Then

$$\mathbf{C}_x = \mathbf{E}^T \mathbf{E} \mathbf{D} \mathbf{E}^T \mathbf{E} = \mathbf{D}$$

is a diagonal matrix.

- 2-D Case $X[u, v] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m, n] K[u, m] K[v, n]$

KLT 缺點: dependent on image
(不實際, 需要一併記錄 transform matrix)

Reference

W. D. Ray and R. M. Driver, “Further decomposition of the Karhunen-Loeve series representation of a stationary random process,” *IEEE Trans. Inf. Theory*, vol. 16, no. 6, pp. 663-668, Nov. 1970.

◎ 7-F Suboptimal Transform-- DCT

- DCT: Discrete Cosine Transform

Suboptimal, but independent of
the input

$$F[u, v] = \frac{2C[u]C[v]}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m, n] \cos \frac{(2m+1)u\pi}{2M} \cos \frac{(2n+1)v\pi}{2N}$$

$$C[0] = 1/\sqrt{2} \quad , C[u] = 1 \text{ for } u \neq 0$$

IDCT: inverse discrete cosine transform

$$f[m, n] = \frac{2}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F[u, v] C[u] C[v] \cos \frac{(2m+1)u\pi}{2M} \cos \frac{(2n+1)v\pi}{2N}$$

對於大部分的影像而言，DCT 能夠近似 KLT (near optimal)

尤其是當 $\text{corr}\{f[m, n], f[m+\tau, n+\eta]\} = \rho^{|\tau|} \rho^{|\eta|}$, $\rho \rightarrow 1$ 時

有 fast algorithm

Advantage: (1) independent of the input (2) near optimal (3) real output

DCT

$$F[u, v] = \frac{2C[u]C[v]}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m, n] \cos \frac{(2m+1)u\pi}{2M} \cos \frac{(2n+1)v\pi}{2N}$$

$$C[0] = 1/\sqrt{2} \quad , C[u] = 1 \text{ for } u \neq 0$$

$[u, v] = [0, 0]$: DC term $F[0, 0] = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m, n]$

$u \neq 0$ or $v \neq 0$: AC terms 借用電路學的名詞

It can also be rewritten as

$$F_1[u, n] = \sqrt{\frac{2}{M}} C[u] \sum_{m=0}^{M-1} f[m, n] \cos \frac{(2m+1)u\pi}{2M} \quad n = 0, 1, 2, \dots, N-1,$$

(N times of M-point 1D DCT)

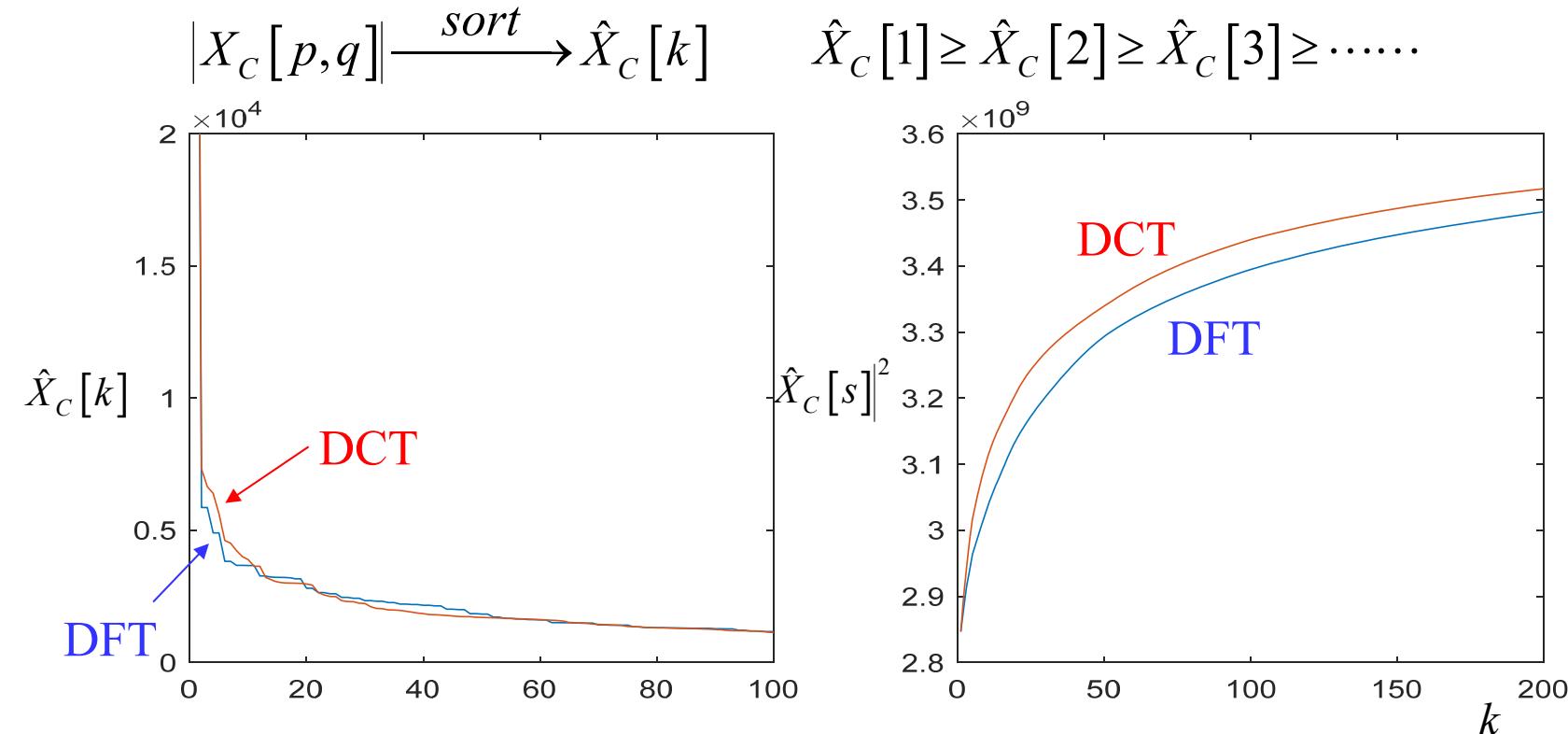
$$F[u, v] = \sqrt{\frac{2}{N}} C[v] \sum_{n=0}^{N-1} F_1[u, n] \cos \frac{(2n+1)v\pi}{2N} \quad u = 0, 1, 2, \dots, M-1,$$

(M times of N-point 1D DCT)

左圖：將 DFT，DCT 各點能量(開根號)由大到小排序

右圖：累積能量

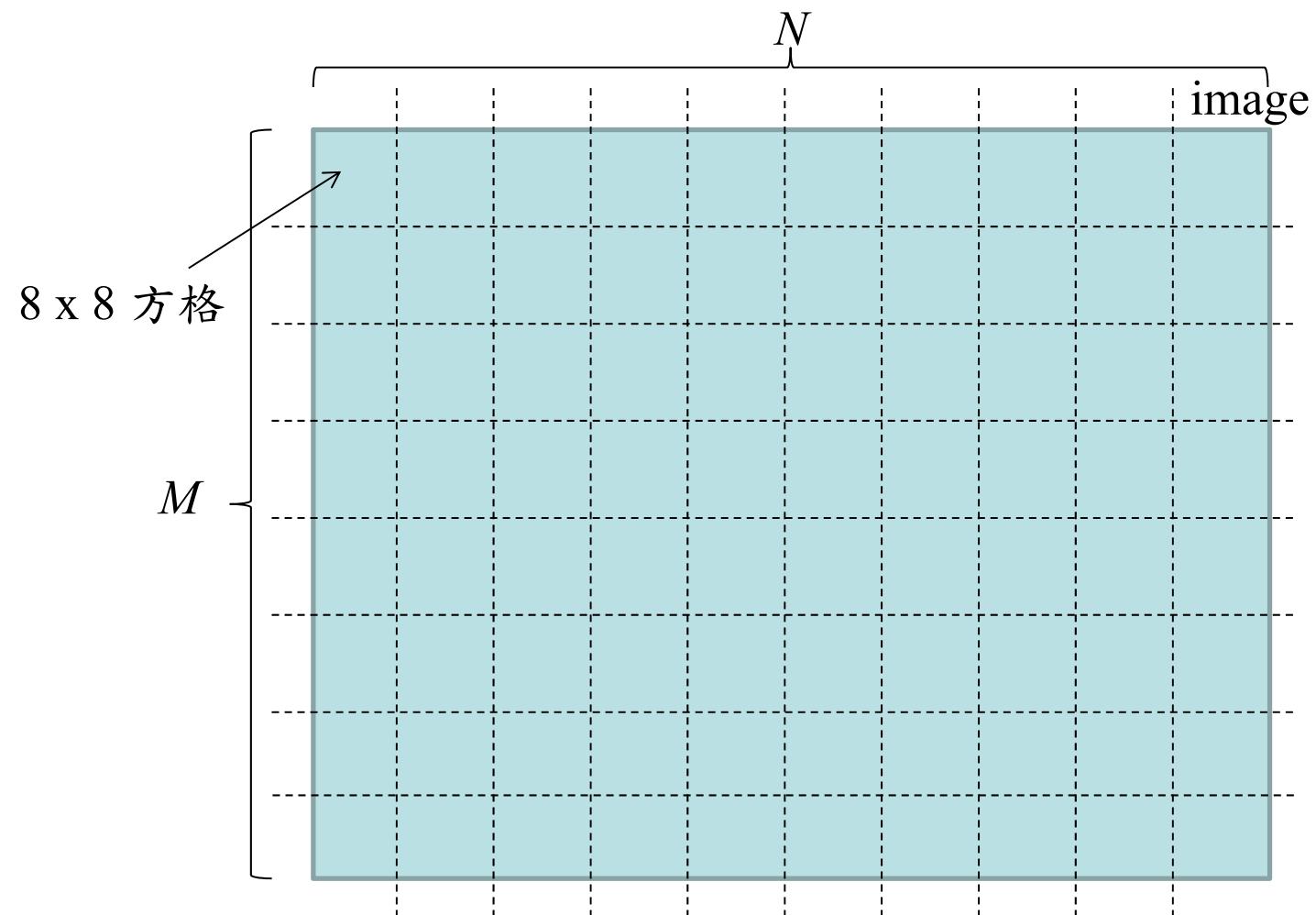
DCT output



Energy concentration at low frequencies: KLT > DCT > DFT

通常，我們將影像切成 8×8 的方格作DCT

Why:



References

- [1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90-93, Jan 1974.
- [2] K. R. Rao and P. Yip, *Discrete Cosine Transform, Algorithms, Advantage, Applications*, New York: Academic, 1990.

附錄十：符合人類知覺的相似度測量工具：結構相似度 Structural Similarity (SSIM)

傳統量測兩個信號 (including images, videos, and vocal signals) 之間相似度的方式：

$$(1) \text{ maximal error } \ Max(|y[m,n] - x[m,n]|)$$

$$(2) \text{ mean square error (MSE) } \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |y[m,n] - x[m,n]|^2$$

$$(3) \text{ normalized mean square error (NMSE) } \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |y[m,n] - x[m,n]|^2}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |x[m,n]|^2}$$

$$(4) \text{ normalized root mean square error (NRMSE) } \sqrt{\frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |y[m,n] - x[m,n]|^2}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |x[m,n]|^2}}$$

(5) L_α -Norm

$$\|y - x\|_\alpha = \left(\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |y[m, n] - x[m, n]|^\alpha \right)^{1/\alpha}$$

$$\frac{1}{MN} \|y - x\|_\alpha = \frac{1}{MN} \left(\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |y[m, n] - x[m, n]|^\alpha \right)^{1/\alpha}$$

(6) signal to noise ratio (SNR), 信號處理常用

$$10 \log_{10} \left(\frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |x[m, n]|^2}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |y[m, n] - x[m, n]|^2} \right)$$

(7) peak signal to noise ratio (PSNR), 影像處理常用

$$10 \log_{10} \left(\frac{X_{Max}^2}{\frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |y[m, n] - x[m, n]|^2} \right)$$

X_{Max} : the maximal possible value of $x[m, n]$

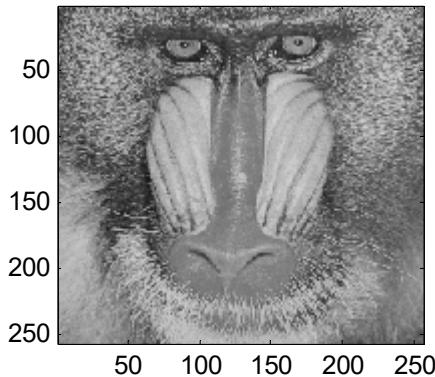
In image processing, $X_{Max} = 255$

for color image: $10 \log_{10} \left(\frac{X_{Max}^2}{\frac{1}{3MN} \sum_{R,G,B} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |y_{color}[m, n] - x_{color}[m, n]|^2} \right)$

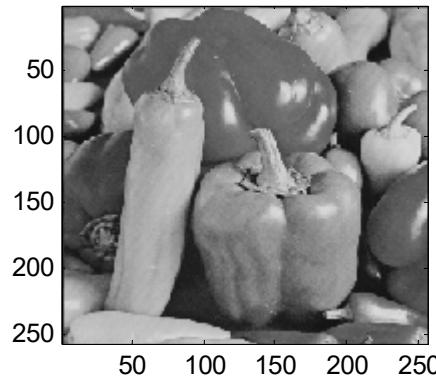
color = R, G, or B

然而，MSE 和 NRMSE 雖然在理論上是合理的，但卻無法反映出實際上兩個影像之間的相似度

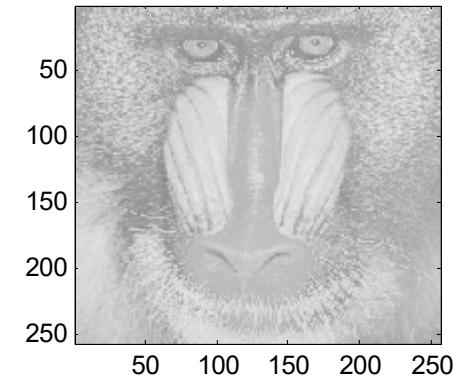
例如：以下這三張圖



圖一



圖二



圖三

$$\text{圖三} = \text{圖一} \times 0.5 + 255.5 \times 0.5$$

照理來說，圖一和圖三較相近

然而，圖一和圖二之間的 NRMSE 為 0.4411

圖一和圖三之間的 NRMSE 為 0.4460

(8) Structural Similarity (SSIM)

有鑑於 MSE 和 PSNR 無法完全反映人類視覺上所感受的誤差，在 2004 年被提出來的新的誤差測量方法

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + (c_1L)^2)}{(\mu_x^2 + \mu_y^2 + (c_1L)^2)} \frac{(2\sigma_{xy} + (c_2L)^2)}{(\sigma_x^2 + \sigma_y^2 + (c_2L)^2)}$$

$$DSSIM(x, y) = 1 - SSIM(x, y)$$

μ_x, μ_y : means of x and y σ_x^2, σ_y^2 : variances of x and y

σ_{xy} : covariance of x and y c_1, c_2 : adjustable constants

$$\sigma_{xy} = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N (x[m, n] - \mu_x)(y[m, n] - \mu_y) \quad \text{where } \mu_x = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N x[m, n]$$

L : the maximal possible value of x – the minimal possible value of x

若使用 SSIM，且前頁的 c_1, c_2 皆選為 $1/\sqrt{L}$

圖一、圖二之間的 SSIM 為 0.1040

圖一、圖三之間的 SSIM 為 0.7720

反映出了圖一、圖三之間確實有很高的相似度

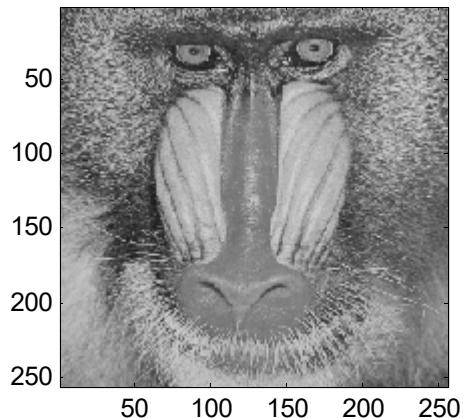
比較：機率上關於相關度 (correlation) 的定義

$$\text{corr}(x, y) = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

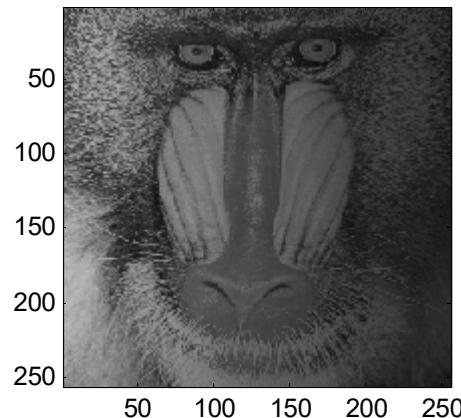
Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

其他幾個用 MSE 和 NRMSE 無法看出相似度，但是可以用 SSIM 看出相似度的情形

影子 shadow



圖四

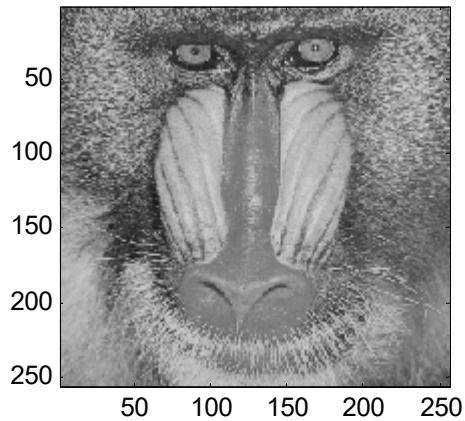


圖五

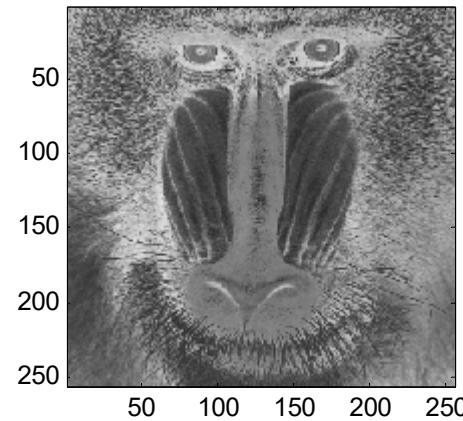
$\text{NRMSE} = 0.4521$ (大於圖一、圖二之間的 NRMSE)

$\text{SSIM} = 0.6010$

底片 the negative of a photo



圖六



圖七

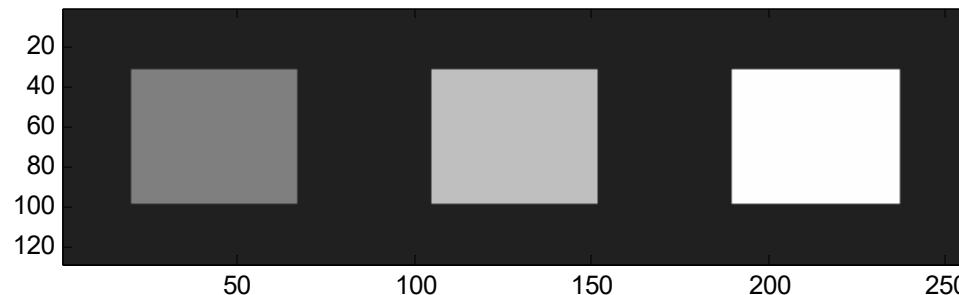
圖七 = 255 – 圖六

NRMSE = 0.5616 (大於圖一、圖二之間的 NRMSE)

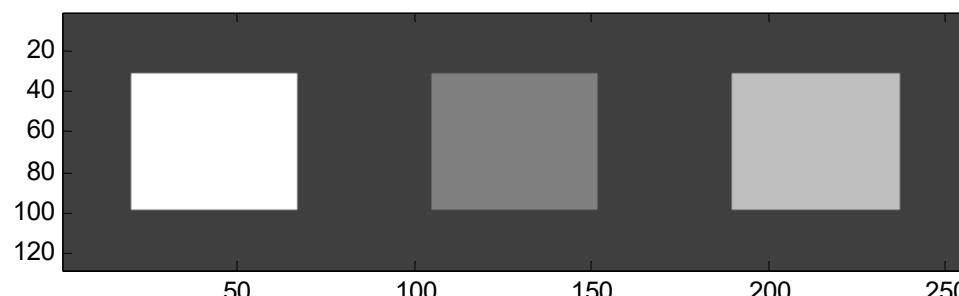
SSIM = -0.8367 (高度負相關)

同形，但亮度不同 (Same shape but different intensity)

圖八



圖九



NRMSE = 0.4978 (大於圖一、圖二之間的 NRMSE)

SSIM = 0.7333

思考：對於 vocal signal (聲音信號而言)

MSE 和 NRMSE 是否真的能反映出兩個信號的相似度？

為什麼？

Perceptual Evaluation of Speech Quality (PESQ)

A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, “Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs,” in *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, pp. 749-752, May 2001.