XIV. Walsh Transform (Hadamard Transform)

◉ 14-A Ideas of Walsh Transforms

- 8-point Walsh transform

$$W[m,n]=\begin{array}{c}\\ \left[\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{array}\right]\end{array}\begin{array}{c} m \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}$$

- Advantages of the Walsh transform:

(1) Real

(2) No multiplication is required

(3) Some properties are similar to those of the DFT

- Forward and inverse Walsh transforms are similar.

forward: $F[m]=\sum_{n=0}^{N-1} f[n]W[m,n]$, inverse: $f[m]=\frac{1}{N}\sum_{n=0}^{N-1} W[m,n]F[n]$

- Alternative names of the Walsh transform:

Hadamard transform, Walsh-Hadamard transform

- Orthogonal Property $\sum_{n=0}^{N-1} W[m_0,n]W[m_1,n]=0$ if $m_0 \neq m_1$ $\sum_{n=0}^{N-1} W[m,n]W[m,n]=N$
- Zero-Crossing Property
- Even / Odd Property
- Fast Algorithm

Useful for spectrum analysis

Sometimes also useful for implementing the convolution

Walsh transform 和 DFT, DCT 有許多相似處

$$W[m,n]=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}, \; DFT[m,n] = \exp(-j2\pi\, m\, n/N),$$

$$\mathbf{DCT} = \begin{bmatrix} 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ 1.3870 & 1.1759 & 0.7857 & 0.2759 & -0.2759 & -0.7857 & -1.1759 & -1.3870 \\ 1.3066 & 0.5412 & -0.5412 & -1.3066 & -1.3066 & -0.5412 & 0.5412 & 1.3066 \\ 1.1759 & -0.2759 & -1.3870 & -0.7857 & 0.7857 & 1.3870 & 0.2759 & -1.1759 \\ 1.0000 & -1.0000 & -1.0000 & 1.0000 & 1.0000 & -1.0000 & -1.0000 & 1.0000 \\ 0.7857 & -1.3870 & 0.2759 & 1.1759 & -1.1759 & -0.2759 & 1.3870 & -0.7857 \\ 0.5412 & -1.3066 & 1.3066 & -0.5412 & -0.5412 & 1.3066 & -1.3066 & 0.5412 \\ 0.2759 & -0.7857 & 1.1759 & -1.3870 & 1.3870 & -1.1759 & 0.7857 & -0.2759 \end{bmatrix}$$

References for Walsh Transforms

[1] K. G. Beanchamp, *Walsh Functions and Their Applications*, Academic Press, New York, 1975.

[2] B. I. Golubov, A. Efimov, and V. Skvortsov, *Walsh Series and Transforms: Theory and Applications*, Kluwer Academic Publishers, Boston, 1991.

[3] H. F. Harmuth, "Applications of Walsh functions in communications," *IEEE Spectrum*, vol. 6, no. 11, pp. 82-91, Nov. 1969.

[4] H. F. Harmuth, *Transmission of Information by Orthogonal Functions*, Springer-Verlag, New York, 1972.

◉ 14-B Generate the Walsh Transform

2-point Walsh transform

$$\mathbf{W}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

4-point Walsh transform

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

How do we obtain the 2^{k+1}-point Walsh transform from the 2^k-point Walsh transform ?

Step 1 $$\mathbf{V}_{2^{k+1}} = \begin{bmatrix} \mathbf{W}_{2^k} & \mathbf{W}_{2^k} \\ \mathbf{W}_{2^k} & -\mathbf{W}_{2^k} \end{bmatrix}$$

Step 2 根據 sign changes 將 rows 的順序重新排列

$$\mathbf{V}_{2^{k+1}} \xrightarrow{\textit{permutation}} \mathbf{W}_{2^{k+1}}$$

已知 $\mathbf{W}_{2^k}$ 每個 row 的 sign change 數，由上到下分別為

$0, 1, 2, 3, \ldots\ldots, 2^k-1$

則 $\mathbf{V}_{2^{k+1}}$ 每個 row 的 sign change 數，由上到下分別為

$0, 3, 4, 7, \ldots\ldots, 2^{k+1}-1, 1, 2, 5, 6, \ldots\ldots, 2^{k+1}-2,$

若 row 的index 由0 開始

則 $\mathbf{V}_{2^{k+1}}$ 第 n 個 row (n is even and $n < N/2$) 的 sign change 為 $2n$

(n is odd and $n < N/2$) 的 sign change 為 $2n + 1$

(n is even and $n \geq N/2$) 的 sign change 為 $2n+1-N$

(n is odd and $n \geq N/2$) 的 sign change 為 $2n-N$

要根據 sign change 的數目將 $\mathbf{V}_{2^{k+1}}$ 的 row 重新排列

$$\mathbf{V}_{2^{k+1}} \xrightarrow{\quad permutation \quad} \mathbf{W}_{2^{k+1}}$$

$$\mathbf{W}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \mathbf{V}_4 = \begin{bmatrix} \mathbf{W}_2 & \mathbf{W}_2 \\ \mathbf{W}_2 & -\mathbf{W}_2 \end{bmatrix} = \begin{array}{c} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \end{array} \begin{array}{c} \text{sign changes} \\ 0 \\ 3 \\ 1 \\ 2 \end{array}$$

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad \mathbf{V}_8 = \begin{array}{c} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix} \end{array} \begin{array}{c} \text{sign changes} \\ 0 \\ 3 \\ 4 \\ 7 \\ 1 \\ 2 \\ 5 \\ 6 \end{array}$$

Constraint for the number of points of the Walsh transform:

N must be a power of 2 (2, 4, 8, 16, 32, ……..)

Although in Matlab it is possible to define the $12 \cdot 2^k$ point or the $20 \cdot 2^k$ point Walsh transform, the inverse transform require the floating-point operation.

◉ 14-C Alternative Forms of the Walsh Transform

標準定義

from zero-crossing

- Sequency ordering (i.e., Walsh ordering) ……. using for signal processing
- Dyadic ordering (i.e., Paley ordering) ……….. using for control
- Natural ordering (i.e., Hadamard ordering) ……using for mathematics

Sequency ordering	Dyadic ordering	Natural ordering	$W[m, n]$
←→	(Gray Code) ←→	(Bit Reversal)	
row 0 =	row 0 =	row 0 =	[1, 1, 1, 1, 1, 1, 1, 1]
row 1 =	row 1 =	row 4 =	[1, 1, 1, 1, –1, –1, –1, –1]
row 2 =	row 3 =	row 6 =	[1, 1, –1, –1, –1, –1, 1, 1]
row 3 =	row 2 =	row 2 =	[1, 1, –1, –1, 1, 1, –1, –1]
row 4 =	row 6 =	row 3 =	[1, –1, –1, 1, 1, –1, –1, 1]
row 5 =	row 7 =	row 7 =	[1, –1, –1, 1, –1, 1, 1, –1]
row 6 =	row 5 =	row 5 =	[1, –1, 1, –1, –1, 1, –1, 1]
row 7 =	row 4 =	row 1 =	[1, –1, 1, –1, 1, –1, 1, –1]

- Dyadic ordering Walsh transform

$$W[m,n]=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&-1&-1&-1&-1\\1&1&-1&-1&1&1&-1&-1\\1&1&-1&-1&-1&-1&1&1\\1&-1&1&-1&1&-1&1&-1\\1&-1&1&-1&-1&1&-1&1\\1&-1&-1&1&1&-1&-1&1\\1&-1&-1&1&-1&1&1&-1\end{bmatrix}$$

- Natural ordering Walsh transform

$$W[m,n]=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&-1&1&-1&1&-1&1&-1\\1&1&-1&-1&1&1&-1&-1\\1&-1&-1&1&1&-1&-1&1\\1&1&1&1&-1&-1&-1&-1\\1&-1&1&-1&-1&1&-1&1\\1&1&-1&-1&-1&-1&1&1\\1&-1&-1&1&-1&1&1&-1\end{bmatrix}$$

- binary code $n = \sum_{p=1}^{k} b_p 2^{p-1}$ to gray code

  When $N = 2^k$

  $g_k = b_k$, $g_q = \mathrm{XOR}(b_{q+1}, b_q)$ for $q = k-1, k-2, \ldots, 1$ $\qquad m = \sum_{q=1}^{k} g_q 2^{q-1}$

- gray code to binary code

  When $N = 2^k$

  $b_k = g_k$, $b_q = \mathrm{XOR}(b_{q+1}, g_q)$ for $q = k-1, k-2, \ldots, 1$

◉ 14-D Properties

(1) Orthogonal Property

(2) Zero-Crossing Property

(3) Even / Odd Property

(4) Linear Property

If $f[n] \Rightarrow F[m]$, $g[n] \Rightarrow G[m]$, ($\Rightarrow$ means the Walsh transform)

then $a\, f[n] + b\, g[n] \Rightarrow a\, F[m] + b\, G[m]$

(5) Addition Property

$$W[m,n]\cdot W[l,n]=W[m\oplus l,n]$$

註： Addition modulo 2 (denoted by $\oplus$)

$0 \oplus 0 = 1 \oplus 1 = 0,\quad 0 \oplus 1 = 1 \oplus 0 = 1,$

$$(\sum_{p=0}^{k} a_k 2^p)\oplus(\sum_{p=0}^{k} b_k 2^p)=\sum_{p=0}^{k}(a_k \oplus b_k)2^p$$

Example:

$$\begin{array}{rr} & 3 \quad\quad 0\ 1\ 1 \\ \oplus & 7 \quad\quad 1\ 1\ 1 \\ \hline & 4 \quad\quad 1\ 0\ 0 \end{array}$$

, therefore $3 \oplus 7 = 4$

$\oplus$: logic addition
(similar to XOR)

(6) Special functions

$\delta[n] = 1$ when $n = 0$, $\delta[n] = 0$ when $n \neq 0$

$\delta[n] \Rightarrow 1, \quad 1 \Rightarrow N\cdot\delta[n]$

(7) Shifting property

If $f[n] \Rightarrow F[m]$, then $f[n \oplus k] \Rightarrow W(k, m)\cdot F[m]$

(8) Modulation property

If $f[n] \Rightarrow F[m]$, then $W(k, n)\cdot f[n] \Rightarrow F[m \oplus k]$

(9) Parseval's Theorem

If $f[n] \Rightarrow F[m]$,

$$\sum_{n=0}^{N-1} \left|f[n]\right|^2 = \frac{1}{N}\sum_{n=0}^{N-1} \left|F[m]\right|^2 ,$$

If $f[n] \Rightarrow F[m]$, $g[n] \Rightarrow G[m]$,

$$\sum_{n=0}^{N-1} f[n]g[n] = \frac{1}{N}\sum_{n=0}^{N-1} F[m]G[m]$$

(10) Convolution Property

If $f[n] \Rightarrow F[m]$, $\quad g[n] \Rightarrow G[m]$,

then $h[n] = f[n] \star g[n] \Rightarrow F[m]\, G[m]$

$\star$ means the "logical convolution"

$$f[n] \star g[n] = W^{-1}\left(W\left(f[n]\right)W\left(g[n]\right)\right)$$

W : Walsh transform

W^{-1} : inverse Walsh transform

$$h[n] = f[n] \star g[n] = \sum_{l=0}^{N-1} f[l]\, g[n \oplus l] = \sum_{l=0}^{N-1} f[n \oplus l]\, g[l]$$

For example, when $N = 8$,

$$h[3] = f[0]g[3] + f[1]g[2] + f[2]g[1] + f[3]g[0] + f[4]g[7] + f[5]g[6] + f[6]g[5] + f[7]g[4]$$

$$h[2] = f[0]g[2] + f[1]g[3] + f[2]g[0] + f[3]g[1] + f[4]g[6] + f[5]g[7] + f[6]g[4] + f[7]g[5]$$

Comparison：In digital signal processing, we often use

linear convolution (standard form of convolution)

$$\sum_{l=0}^{N-1} f[l]g[n-l]$$

circular convolution

$$\sum_{l=0}^{N-1} f[l]g[((n-l))_N]$$

$$IDFT_N\{DFT_N[f[n]]DFT_N[g[n]]\} = \sum_{l=0}^{N-1} f[l]g[((n-l))_N]$$

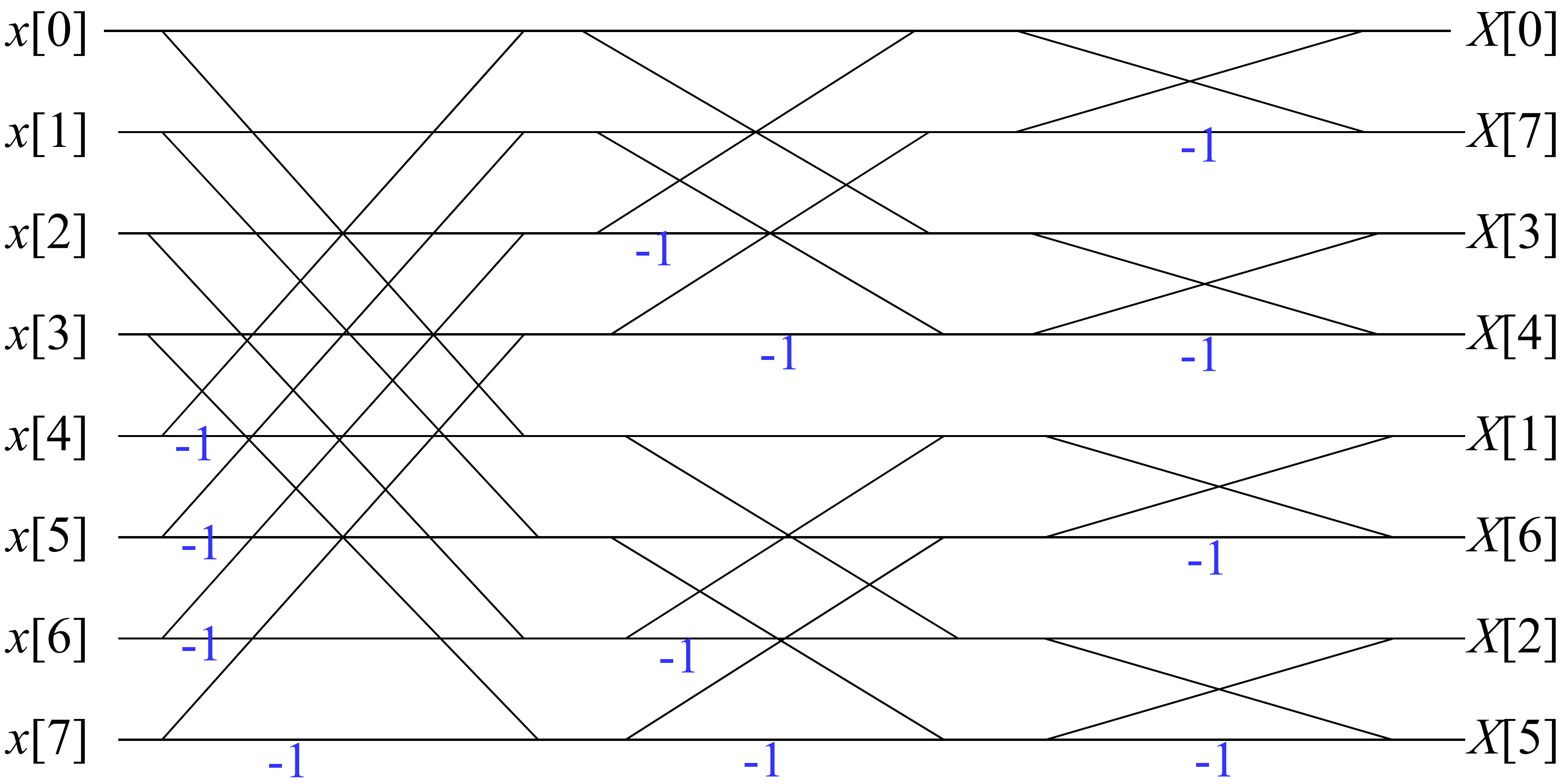
For example, when $N = 8$,

$$H[3] = f[0]g[3] + f[1]g[2] + f[2]g[1] + f[3]g[0] + f[4]g[7] + f[5]g[6] + f[6]g[5] + f[7]g[4]$$

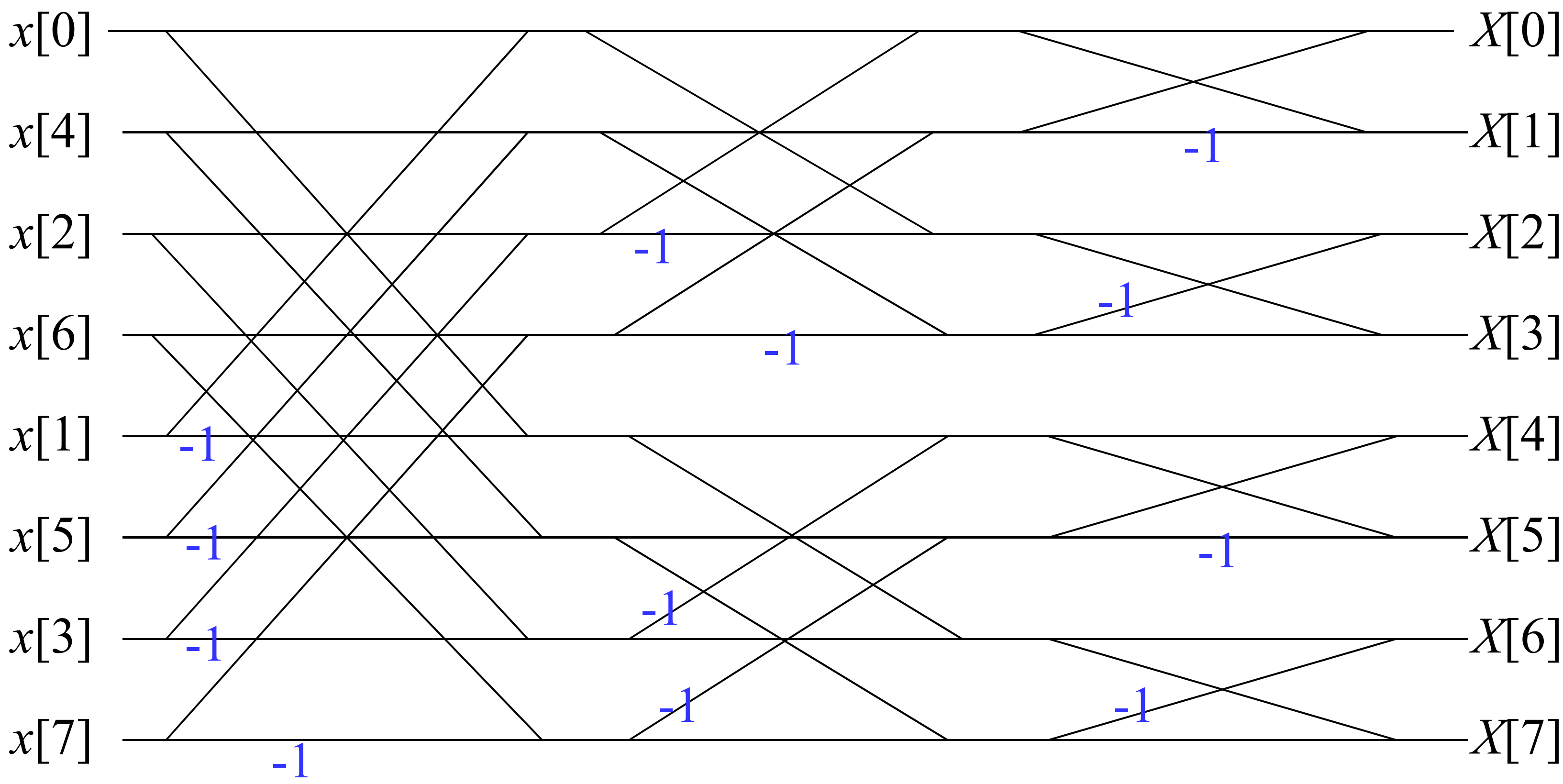
$$H[2] = f[0]g[2] + f[1]g[1] + f[2]g[0] + f[3]g[7] + f[4]g[6] + f[5]g[5] + f[6]g[4] + f[7]g[3]$$

⦿ 14-E Butterfly Fast Algorithm

(Method 1) John L. Shark's Algorithm



(Method 2) Manz's Sequence Algorithm



There are other fast implementation algorithm for the Walsh transform.

◎ 14-F Applications

Walsh transform 適合作 spectrum analysis，但未必適合作convolution

↓

may not be better than DFT, DCT

Applications of the Walsh transform

Bandwidth reduction

High resolution

Modulation and Multiplexing

Information coding

Feature extraction

ECG signal (in medical signal processing) analysis

Hadamard spectrometer

Avoiding quantization error

- The Walsh transform is suitable for the function that is a combination of Step functions

New Applications: CDMA (code division multiple access)

◎ 14-G Jacket Transform

把部分的 1 用 $\pm 2^k$ 取代

4-point Jacket transform

$$\mathbf{J}_4 = \begin{bmatrix} 1 & x & x & 1 \\ 1 & w & -w & -1 \\ 1 & -x & -x & 1 \\ 1 & -w & w & 1 \end{bmatrix} \qquad w = 2^k, \quad x = 2^h,$$

2^{k+1}-point Jacket

$$\mathbf{J}_{2^{k+1}} = \mathbf{P}\begin{bmatrix} \mathbf{J}_{2^k} & \mathbf{J}_{2^k} \\ \mathbf{J}_{2^k} & -\mathbf{J}_{2^k} \end{bmatrix}$$

P: row permutation

[Ref] M. H. Lee, “A new reverse Jacket transform and its fast algorithm,” *IEEE Trans. Circuits Syst.-II* , vol. 47, pp. 39-46, 2000.

◉ 14-H Haar Transform

$\mathbf{y} = \mathbf{H}\mathbf{x}$

$N = 2$ $\quad \mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ $\qquad N = 4$ $\quad \mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$

$N = 8$ $\quad \mathbf{H}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$

[Ref] H. F. Harmuth, *Transmission of Information by Orthogonal Functions*, Springer-Verlag, New York, 1972

$N = 16$

$$\mathbf{H}_{16} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$H[m, n]$ 的值 ($m = 0, 1, \ldots, 2^k-1$, $n = 0, 1, \ldots, 2^k-1$)：

$$H[0, n] = 1 \text{ for all } n$$

If $2^h \leq m < 2^{h+1}$

$$H[m, n] = 1 \quad \text{for } (m-2^h)2^{k-h} \leq n < (m-2^h+1/2)2^{k-h}$$

$$H[m, n] = -1 \quad \text{for } (m-2^h+1/2)2^{k-h} \leq n < (m-2^h+1)2\mathrm{k}^{-h}$$

$$H[m, n] = 0 \quad \text{otherwise}$$

運算量比 Walsh transforms 更少

Applications: localized spectrum analysis, edge detection

Transforms	Running Time	terms required for NRMSE $< 10^{-5}$
DFT	9.5 sec	43
Walsh Transform	2.2 sec	65
Haar Transform	0.3 sec	128

Main Advantage of the Haar Transform

(1) Fast (but this advantage is no longer important)

(2) Analysis of the local high frequency component
(The wavelet transform is a generalization of the Haar transform)

(3) Extracting local features
(Example: Adaboost face detection)

附錄十六　SCI Papers 查詢方式

我們經常聽到 SCI 論文，impact factor….那麼什麼是 SCI 和 impact factor？什麼樣的論文是 SCI Papers? Impact factor 號如何查詢？

SCI 全名： Science Citation Index

(A) SCI 相關網站：ISI Web of Knowledge

連結至 ISI Web of Knowledge

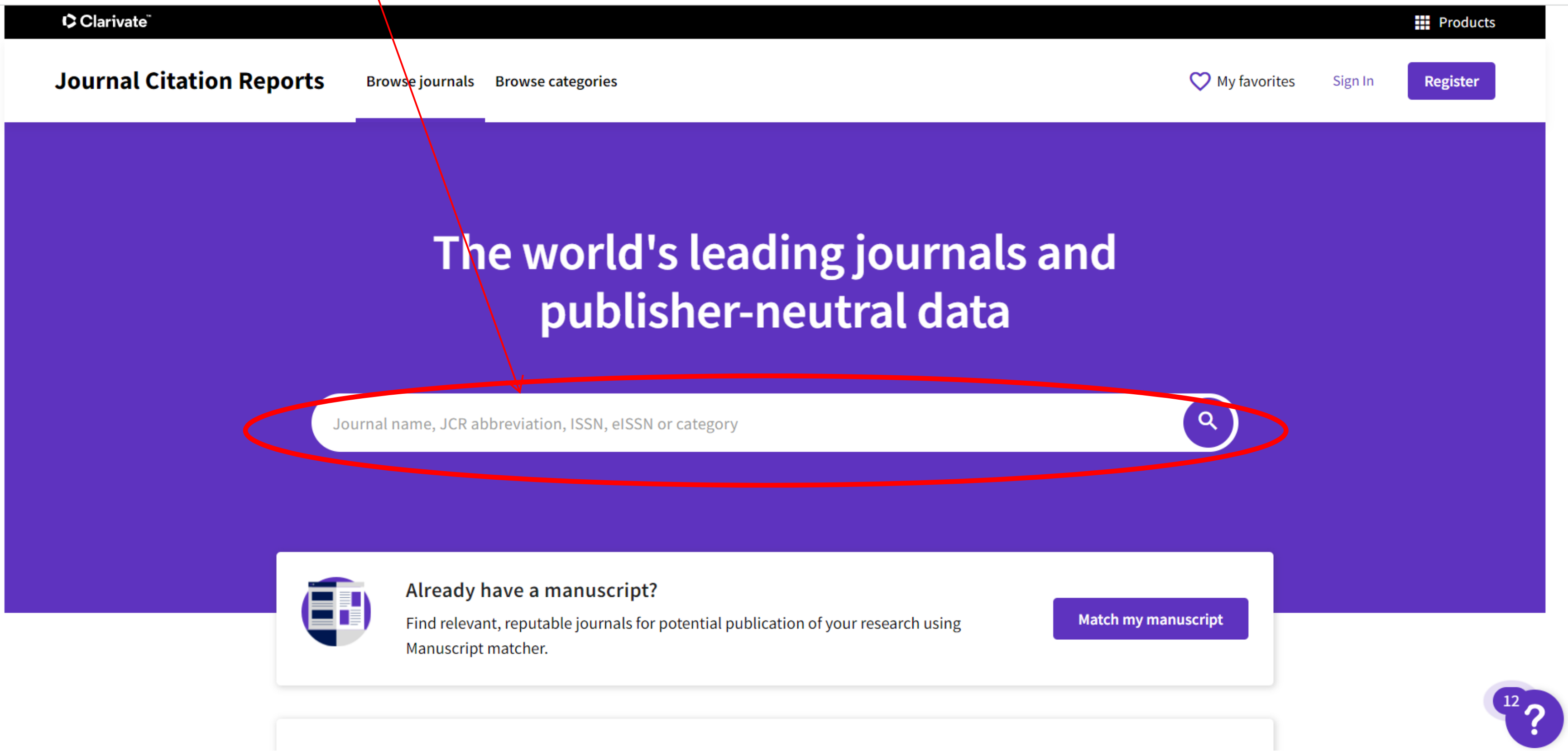
http://admin-apps.webofknowledge.com/JCR/JCR?RQ=HOME

註：必需要在台大上網，或是在其他有付錢給 ISI 的學術單位上網，才可以使用 ISI Web of Knowledge

(B) 在 Go to Journal Profile

輸入你想查詢的期刊 (完整名稱)

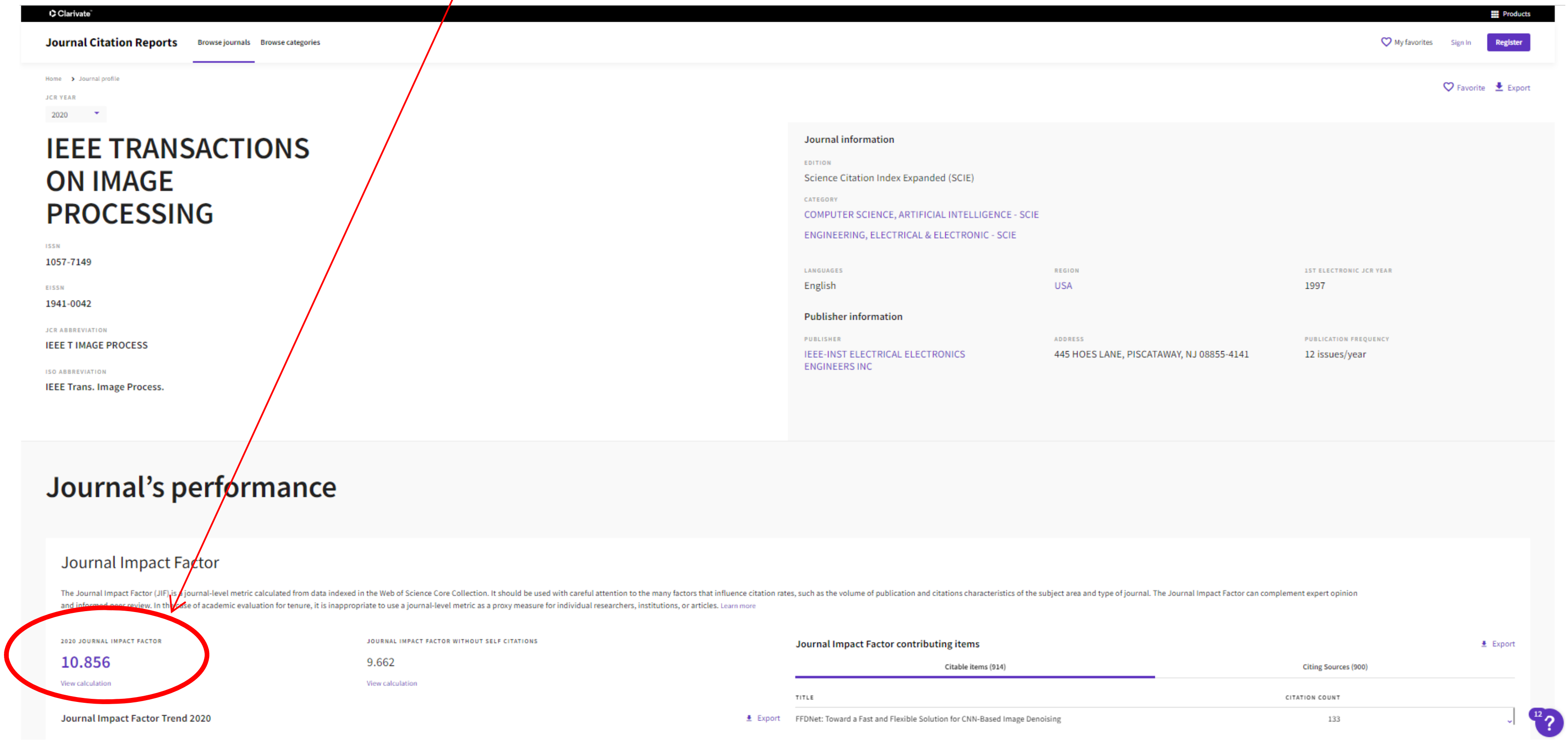
輸入



若有搜尋到，則代表這個期刊是 SCI 期刊

並且會顯示出這個期刊的 impact factor

Impact Factor (影響係數)



(C) 關於 impact factor (影響係數)：

若一個 journal 裡面的文章，被別人引用的次數越多，則這個 journal 的 impact factor 越高

一般而言， impact factor 在 3.5 以上的 journals，已經算是高水準的期刊

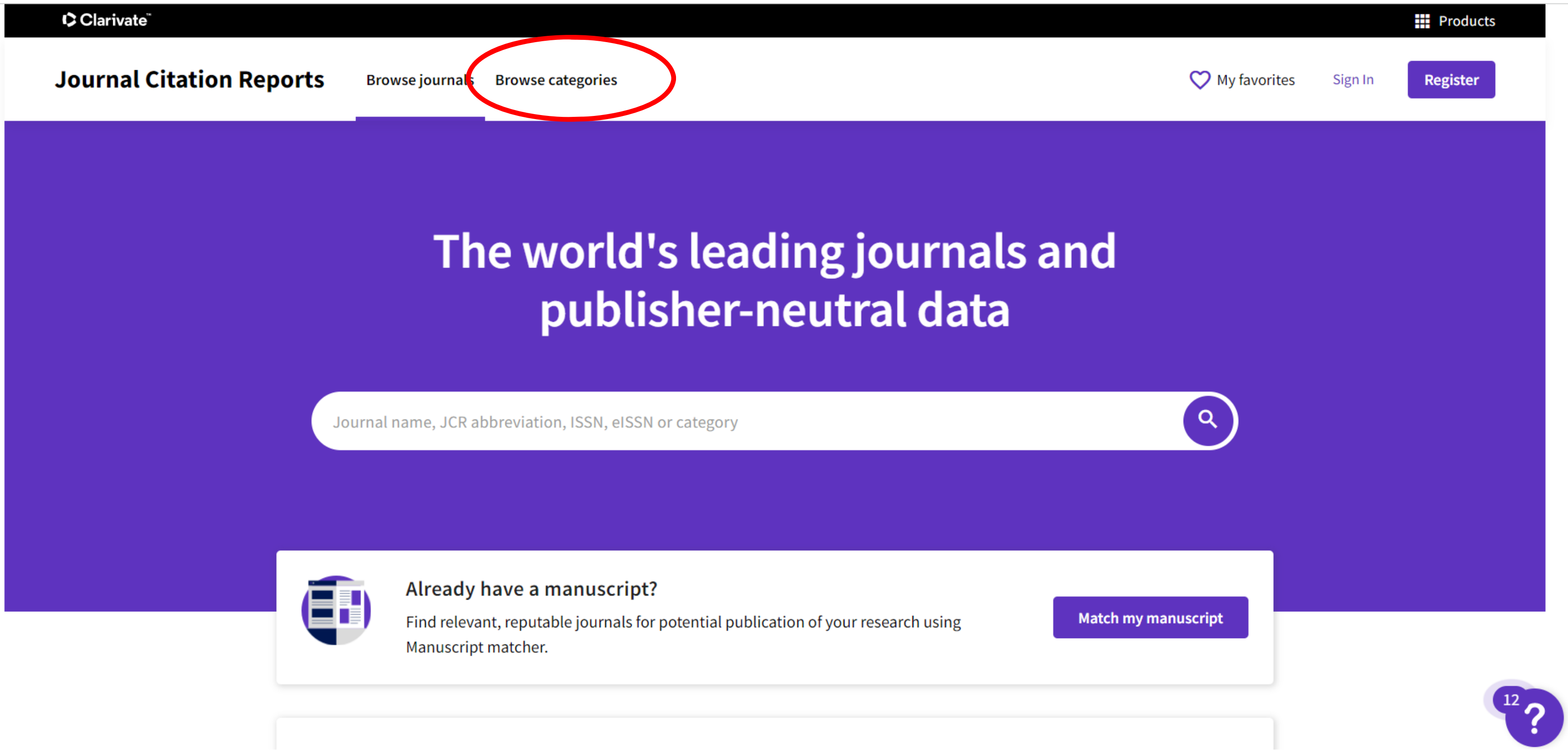
中等水準的期刊的 impact factors 在 1.5 到 3.5 之間

Nature 的 impact factor 為 49.962
Science 的 impact factor 為 47.728

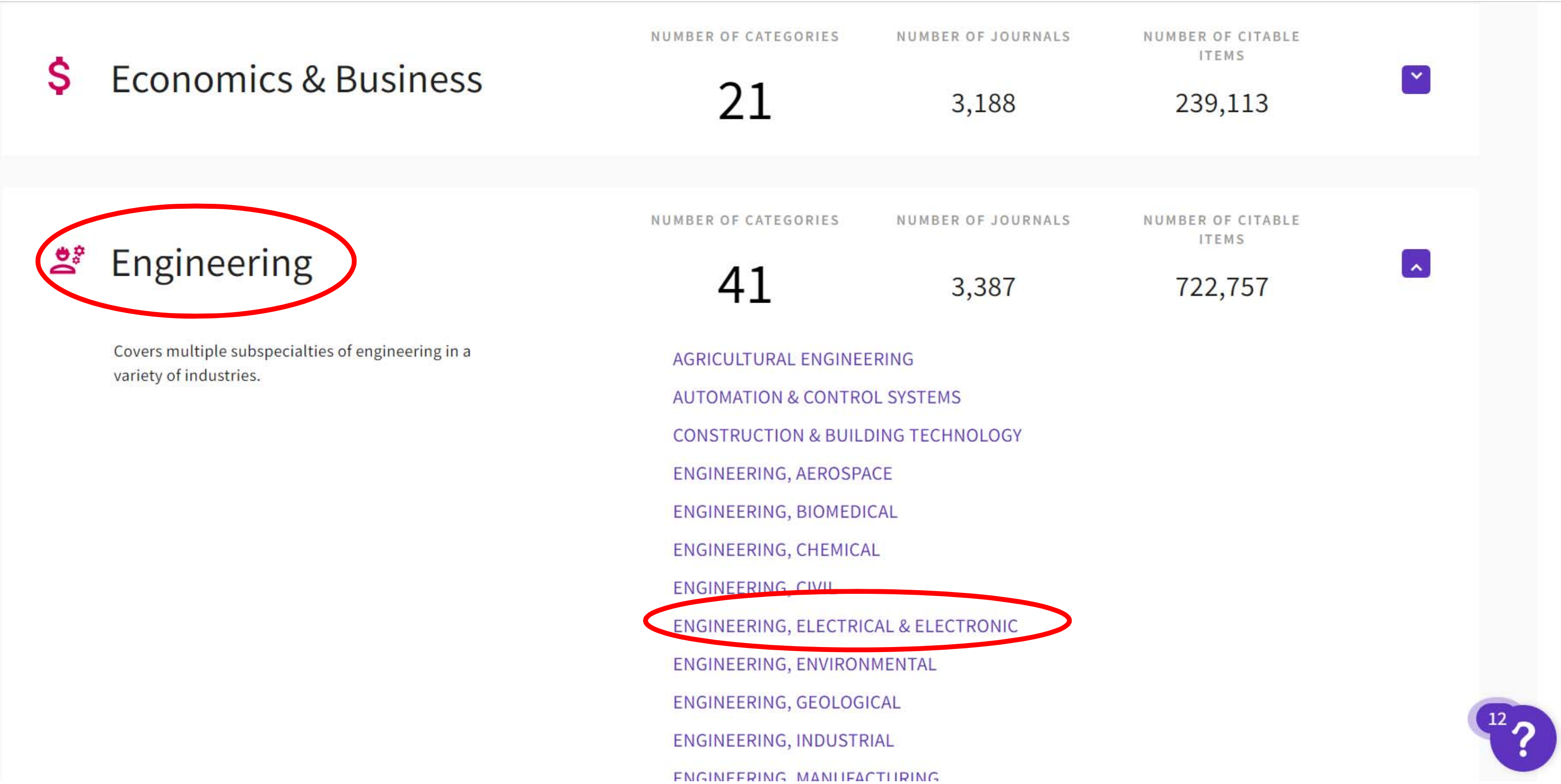
IEEE 系列的期刊的 impact factors 通常在 2 到 13 之間

(D) 要查詢一個領域有哪些 SCI journals

連結至 ISI Web of Knowledge 之後，點選「Browse Category」



再選擇要查詢的 category，如



(E) EI (Engineering Village)

官方網站： www.engineeringvillage.org

http://www.engineeringvillage.com/search/quick.url

查詢期刊或研討會是否為 EI

http://tul.blog.ntu.edu.tw/archives/4627

(F) SSCI (Social Science Citation Index)

比較偏向於社會科學

http://www.thomsonscientific.com/cgi-bin/jrnlst/jloptions.cgi?PC=J

(G) Conference 排名

Microsoft Academic Search 有列出各領域知名的 conferences 並加以排名 (大致上也是被引用越多的排名越前面)

和通訊與信號處理相關的 conferences，大多排名於

http://academic.research.microsoft.com/RankList?entitytype=3&topDomainID=2&subDomainID=0&last=0&start=1&end=100

或

http://academic.research.microsoft.com/RankList?entitytype=3&topDomainID=8&subDomainID=0&last=0&start=1&end=100

(H) H Index

論文除了量以外，也要注意 citation 的次數

將發表的論文的 citation 次數從高到低做排序
如果排名第 N 名的論文 citation 數量大於等於 N
但是排名第 N+1名的論文 citation 數量小於等於 N+1

則 H index = N

Example: 假設有一個學者發表了10篇論文， citation 由多到少分別為
33, 24, 18, 13, 9, 7, 4, 3, 1, 1
則這個學者的 H-index 為 6

寫論文和投稿的經驗談

研究生經常會寫論文並且投稿。要如何讓論文投稿之後能夠順利的被接受，相信是同學們所期盼的，畢竟每篇論文都是大家花了不少時間的心血結晶，若論文能夠順利的被接受，也代表了自己的成果總算獲得了肯定。然而，影響論文是否被接受的因素很多，一個好的研究成果，還是配合好的編寫技巧，才可以被一流的期刊或研討會所接受。以下是個人關於論文編寫與投稿的經驗談：

(1) 你的論文的「賣點」(優點) 是什麼？人家有沒有辦法一眼看得出來你論文的「賣點」？

寫論文其實就是在推銷商品，而所謂的「商品」，就是你的「研究成果」。要說服人家接受你的商品，首先就是要強調你的商品的「賣點」。

(2) 和既有的方法的比較是否足夠？

要證明你所提出的方法是有效的，最好的方式，就是和既有的方法相比較，而且比較的對象越多越好，越新越好。

(3) 和前人的方法相比，你的方法創新的地方在何處？審稿者是否能看得出來你論文創新的地方？

(4) 就算你的文章和理論相關，最好也多提出實際應用的例子

(5) 參考資料越多越好，越新越好
(在研究一個領域時，論文 survey 的量要足夠)

(6) Previous work (前人已經提出的概念) 精簡介紹即可，多強調自己的貢獻。Introduction 加上 Previous work 最好不要超過一篇論文的四分之一

(7) 英文表達能力要有一定的水準

(8)可以多用數學式和圖來解釋概念，有時會比文字還清楚

通常東方人英文表達能力有限。審稿者經常會看你們的圖表和數學式(而非文字) 來判斷你們論文的品質

(9) 同樣的道理，可以用「條列式」的方式來取代一大段文字來描述方法的觀念、流程、或優點

(10) 可以用 Conference 的期限來要求自己多寫研討會論文，之後再陸續改成期刊論文投稿，如此一年的論文量將很可觀

(11)多注意格式，不同的期刊或研討會，對格式的要求也不同

(12) 最後，問自己一個問題：

如果你是審稿者，你會滿意你寫的這一篇論文嗎？

若答案是肯定的再投稿

XV. Orthogonal Transform and Multiplexing

◉ 15-A Orthogonal and Dual Orthogonal

Any $M \times N$ discrete linear transform can be expressed as the matrix form:

$$\underbrace{\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[M-1] \end{bmatrix}}_{\mathbf{Y}\,=} = \underbrace{\begin{bmatrix} \phi_0^*[0] & \phi_0^*[1] & \phi_0^*[2] & \cdots & \phi_0^*[N-1] \\ \phi_1^*[0] & \phi_1^*[1] & \phi_1^*[2] & \cdots & \phi_1^*[N-1] \\ \phi_2^*[0] & \phi_2^*[1] & \phi_2^*[2] & \cdots & \phi_2^*[N-1] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{M-1}^*[0] & \phi_{M-1}^*[1] & \phi_{M-1}^*[2] & \cdots & \phi_{M-1}^*[N-1] \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix}}_{\mathbf{X}}$$

$$y[m] = \langle x[n], \phi_m[n] \rangle = \sum_{n=0}^{N-1} x[n]\phi_m^*[n]$$

inner product

Orthogonal: $\langle \phi_k[n], \phi_h[n] \rangle = \sum_{n=0}^{N-1} \phi_k[n]\phi_h^*[n] = 0$ **when** $\boldsymbol{k \neq h}$

orthogonal transforms 的例子：

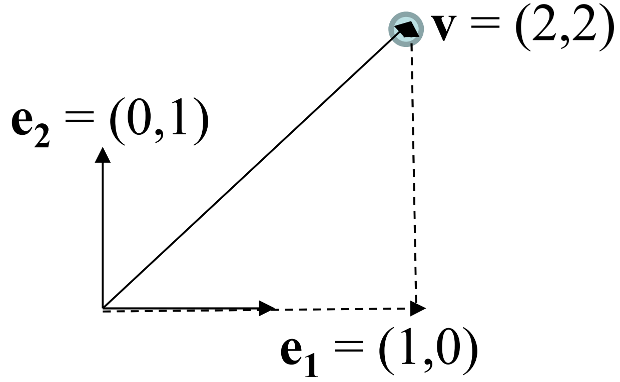
- discrete Fourier transform
- discrete cosine, sine, Hartley transforms
- Walsh Transform, Haar Transform
- discrete Legendre transform
- discrete orthogonal polynomial transforms

  Hahn, Meixner, Krawtchouk, Charlier

為什麼在信號處理上，我們經常用 orthogonal transform?

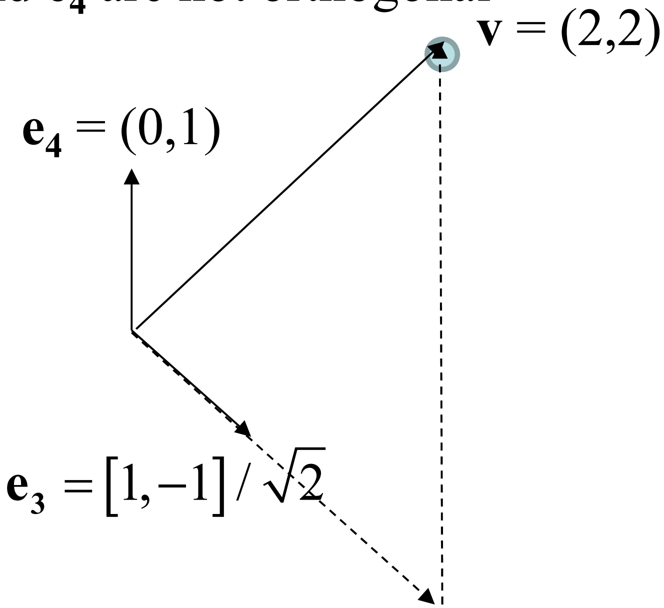
Orthogonal transform 最大的好處何在？

$\mathbf{e_1}$ and $\mathbf{e_2}$ are orthogonal



$$\mathbf{v} = 2\mathbf{e_1} + 2\mathbf{e_2}$$

$\mathbf{e_3}$ and $\mathbf{e_4}$ are not orthogonal



$$\mathbf{v} = 2\sqrt{2}\mathbf{e_3} + 4\mathbf{e_4}$$

- If partial terms are used for reconstruction

for orthogonal case,

perfect reconstruction: $x[n]=\sum_{m=0}^{N-1} C_m^{-1} y[m]\phi_m[n]$

partial reconstruction: $x_K[n]=\sum_{m=0}^{K-1} C_m^{-1} y[m]\phi_m[n] \qquad K<N$

reconstruction error of partial reconstruction

$$\left\|x[n]-x_K[n]\right\|^2=\sum_{n=0}^{N-1}\left\|\sum_{m=K}^{N-1} C_m^{-1} y[m]\phi_m[n]\right\|^2$$

$$=\sum_{n=0}^{N-1}\sum_{m=K}^{N-1} C_m^{-1} y[m]\phi_m[n]\sum_{m_1=K}^{N-1} C_{m_1}^{-1} y^*[m_1]\phi_{m_1}^*[n]$$

$$=\sum_{m=K}^{N-1}\sum_{m_1=K}^{N-1} C_m^{-1} y[m]C_{m_1}^{-1} y^*[m_1]\sum_{n=0}^{N-1}\phi_m[n]\phi_{m_1}^*[n]$$

$$=\sum_{m=K}^{N-1}\sum_{m_1=K}^{N-1} C_m^{-1} y[m]C_{m_1}^{-1} y^*[m_1]C_m\delta[m-m_1]=\sum_{m=K}^{N-1} C_m^{-1}\left|y[m]\right|^2$$

由於 $C_m^{-1}\left|y[m]\right|^2$ 一定是正的，可以保證 K 越大, reconstruction error 越小

For non-orthogonal case,

perfect reconstruction: $x[n] = \sum_{m=0}^{N-1} B[n,m]\,y[m]$ $\quad \mathbf{B} = \mathbf{A}^{-1}$

partial reconstruction: $x_K[n] = \sum_{m=0}^{K-1} B[n,m]\,y[m]$ $\quad K < N$

reconstruction error of partial reconstruction

$$\left\| x[n] - x_K[n] \right\|^2 = \sum_{n=0}^{N-1} \left\| \sum_{m=K}^{N-1} B[n,m]\,y[m] \right\|^2$$

$$= \sum_{n=0}^{N-1} \sum_{m=K}^{N-1} B[n,m]\,y[m] \sum_{m_1=K}^{N-1} B^*[n,m_1]\,y^*[m_1]$$

$$= \sum_{m=K}^{N-1} \sum_{m_1=K}^{N-1} y[m]\,y^*[m_1] \sum_{n=0}^{N-1} B[n,m]\,B^*[n,m_1]$$

由於 $y[m]y^*[m_1]\sum_{n=0}^{N-1} B[n,m]B^*[n,m_1]$ 不一定是正的，

無法保證 K 越大, reconstruction error 越小

◎ 15-B Frequency and Time Division Multiplexing

傳統 Digital Modulation and Multiplexing：使用 Fourier transform

- **Frequency-Division Multiplexing (FDM)**

$$z(t)=\sum_{n=0}^{N-1} X_n \exp\left(j2\pi f_n t\right) \qquad X_n = 0 \text{ or } 1$$

X_n can also be set to be -1 or 1

When (1) $t \in [0, T]$ (2) $f_n = n/T$

$$z(t)=\sum_{n=0}^{N-1} X_n \exp\left(j\frac{2\pi nt}{T}\right)$$

it becomes the **orthogonal frequency-division multiplexing (OFDM) in the continuous case.**

Furthermore, if the time-axis is also sampled

$t = mT/N,\qquad m = 0, 1, 2, \ldots.., N-1$

$t \in [0,T]$
sampling for t-axis

$$z\left(m\frac{T}{N}\right)=\sum_{n=0}^{N-1}X_n \exp\left(j\frac{2\pi nm}{N}\right)$$

(OFDM in the discrete case)

then the OFDM is equivalent to the transform matrix of the inverse discrete Fourier transform (IDFT), which is one of the discrete orthogonal transform.

Modulation: $Y_m = z\left(m\frac{T}{N}\right)=\sum_{m=0}^{N-1}A[m,n]X_n$

$$\mathbf{A}=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{j\frac{2\pi}{N}} & e^{j\frac{4\pi}{N}} & \cdots & e^{j\frac{2(N-1)\pi}{N}} \\ 1 & e^{j\frac{4\pi}{N}} & e^{j\frac{8\pi}{N}} & \cdots & e^{j\frac{4(N-1)\pi}{N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{j\frac{2(N-1)\pi}{N}} & e^{j\frac{4(N-1)\pi}{N}} & \cdots & e^{j\frac{2(N-1)(N-1)\pi}{N}} \end{bmatrix}$$

Modulation: $$Y_m = \sum_{m=0}^{N-1} A[m,n] X_n$$

Demodulation: $$X_n = \frac{1}{N} \sum_{m=0}^{N-1} A^*[m,n] Y_m$$

Example: $N = 8$

$$X_n = [1, 0, 1, 1, 0, 0, 1, 1] \qquad (n = 0 \sim 7)$$

• Time-Division Multiplexing (TDM)

$$z(0)=X_0,\quad z\left(\frac{T}{N}\right)=X_1,\quad z\left(2\frac{T}{N}\right)=X_2,\quad \cdots\cdots,\quad z\left((N-1)\frac{T}{N}\right)=X_{N-1}$$

$$y(m)=z\left(m\frac{T}{N}\right)=\sum_{m=0}^{N-1}A[m,n]X_n$$

$$\mathbf{A}=\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

(also a discrete orthogonal transform)

思考：

既然 time-division multiplexing 那麼簡單

那為什麼要使用 frequency-division multiplexing
和 orthogonal frequency-division multiplexing (OFDM)?

◉ 15-C Code Division Multiple Access (CDMA)

除了 frequency-division multiplexing 和 time-division multiplexing，是否還有其他 multiplexing 的方式？

使用其他的 orthogonal transforms
即 code division multiple access (CDMA)

CDMA is an important topic in spread spectrum communication

參考資料

[1] M. A. Abu-Rgheff, *Introduction to CDMA Wireless Communications*, Academic, London, 2007

[2] 邱國書, 陳立民譯, “CDMA 展頻通訊原理,” 五南, 台北, 2002.

CDMA 最常使用的 orthogonal transform 為 Walsh transform

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \begin{array}{l} \leftarrow \text{channel 1} \\ \leftarrow \text{channel 2} \\ \leftarrow \text{channel 3} \\ \leftarrow \text{channel 4} \\ \leftarrow \text{channel 5} \\ \leftarrow \text{channel 6} \\ \leftarrow \text{channel 7} \\ \leftarrow \text{channel 8} \end{array}$$

channel 1　channel 2

當有兩組人在同一個房間裡交談 (A 和B交談)，(C 和D交談) ，
如何才能夠彼此不互相干擾？

(1) Different Time

(2) Different Tone

(3) Different Language

CDMA 分為：

(1) Orthogonal Type (2) Pseudorandom Sequence Type

Orthogonal Type 的例子： 兩組資料 [1, 0, 1] [1, 1, 0]

(1) 將 0 變為 −1 [1, −1, 1] [1, 1, −1]

(2) 1, −1, 1 modulated by [1, 1, 1, 1, 1, 1, 1, 1] (channel 1)

→ [1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1]

1, 1, −1 modulated by [1, 1, 1, 1, -1, -1, -1, -1] (channel 2)

→ [1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1]

(3) 相合

[2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, -2, -2, -2, -2, 0, 0, 0, 0, 2, 2, 2, 2] demodulation

[2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, -2, -2, -2, -2, 0, 0, 0, 0, 2, 2, 2, 2]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1,1]

內積＝8

注意：

(1) 使用 N-point Walsh transform 時，總共可以有N 個 channels

(2) 除了 Walsh transform 以外，其他的 orthogonal transform 也可以使用

(3) 使用 Walsh transform 的好處

- Orthogonal Transform 共通的問題: 需要同步 synchronization

$\mathbf{R_1} = [\ 1,\ \ 1,\ \ 1,\ \ 1,\ \ 1,\ \ 1,\ \ 1,\ \ 1]$

$\mathbf{R_2} = [\ 1,\ \ 1,\ \ 1,\ \ 1, -1, -1, -1, -1]$

$\mathbf{R_5} = [\ 1, -1, -1,\ \ 1,\ \ 1, -1, -1,\ \ 1]$

$\mathbf{R_8} = [\ 1, -1,\ \ 1, -1,\ \ 1, -1,\ \ 1, -1]$

但是某些 basis, 就算不同步也近似 orthogonal

$<\mathbf{R_1}[n], \mathbf{R_1}[n]> = 8$, $<\mathbf{R_1}[n], \mathbf{R_k}[n]> = 0$ if $k \neq 1$

$<\mathbf{R_1}[n], \mathbf{R_k}[n-1]> = 2$ or 0 if $k \neq 1$.

這裡的shift為circular shift

CDMA 的優點：

(1) 運算量相對於 frequency division multiplexing 減少很多

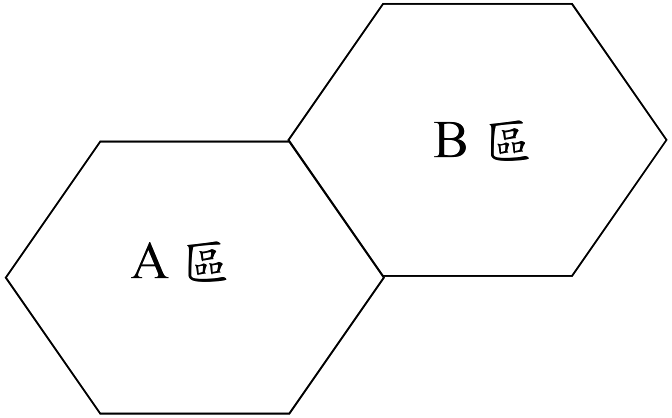
(2) 可以減少 noise 及 interference的影響

(3) 可以應用在保密和安全傳輸上

(4) 就算只接收部分的信號，也有可能把原來的信號 recover 回來

(5) 相鄰的區域的干擾問題可以減少

相鄰的區域，使用差距最大的「語言」，則干擾最少



假設 A 區使用的 orthogonal basis 為 $\phi_k[n]$, $k = 0, 1, 2, \ldots, N-1$

B 區使用的 orthogonal basis 為 $\mu_h[n]$, $h = 0, 1, 2, \ldots, N-1$

設法使 $\max\left(\left|\dfrac{\langle \phi_k[n], \mu_h[n]\rangle}{\langle \phi_k[n], \phi_h[n]\rangle}\right|\right)$ 為最小

$k = 0, 1, 2, \ldots, N-1$, $h = 0, 1, 2, \ldots, N-1$

附錄十七　常用的影像修飾方法

(1) Lightening and Darkening

Input → YCbCr → $Y_o = f(Y)$ → Output

RGB to YCbCr

Cb unchanged
Cr unchanged

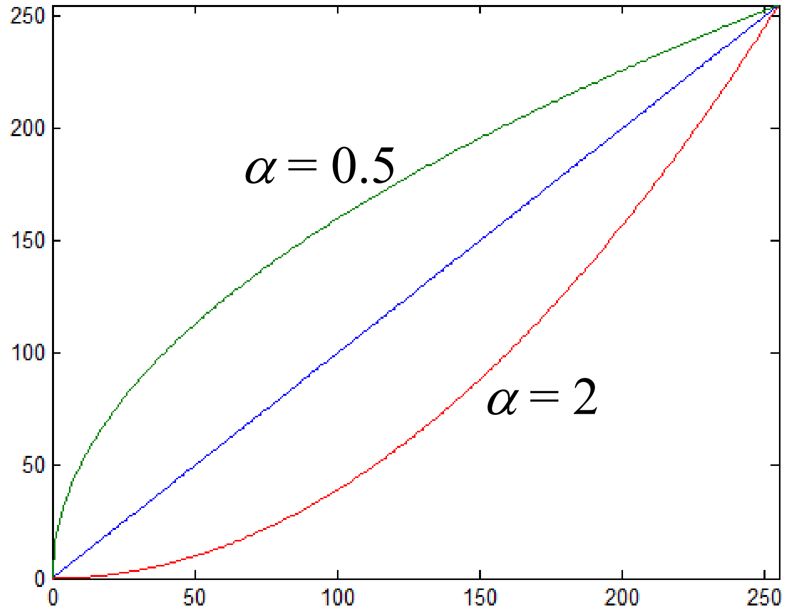
YCbCr to RGB

Example:

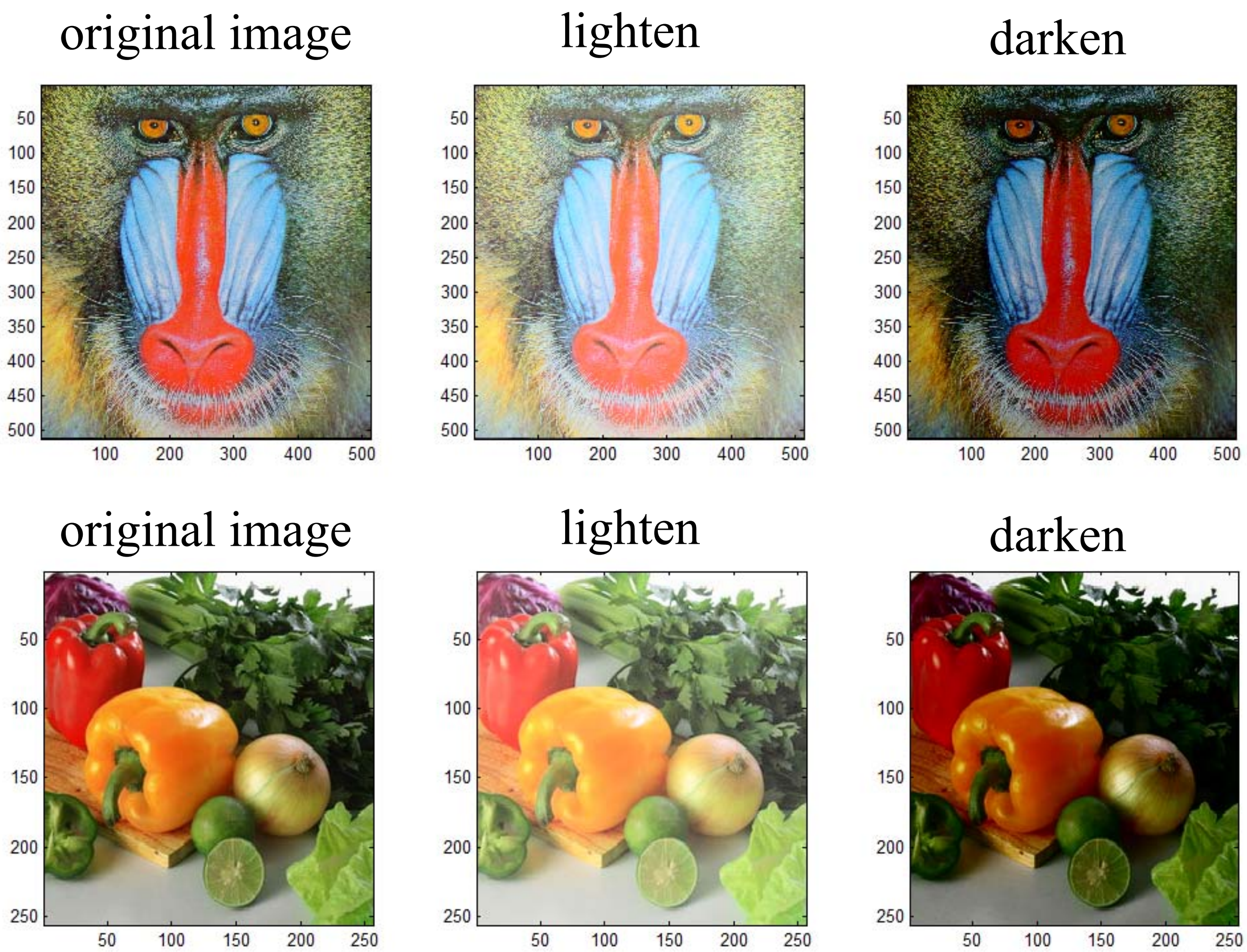
$$f(Y) = 255\left(\frac{Y}{255}\right)^{\alpha}$$

$\alpha < 1$: lightening

$\alpha > 1$: darening



附錄十七　常用的影像修飾方法



附錄十七　常用的影像修飾方法

(2) Morphology

(2-1) Erosion　　(去除區域外圍)

$$A[m,n] = A[m,n] \,\&\, A[m-1,n] \,\&\, A[m+1,n] \,\&\, A[m,n-1] \,\&\, A[m,n+1]$$

*	*	*	*	*	*	*
*	*	*	*	~~O~~	*	*
*	*	~~O~~	~~O~~	O	~~O~~	*
*	~~O~~	O	O	O	~~O~~	*
*	~~O~~	O	O	O	~~O~~	*
*	~~O~~	O	O	O	~~O~~	*
*	*	~~O~~	O	~~O~~	*	*
*	*	*	~~O~~	*	*	*
*	*	*	*	*	*	*

*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	O	*	*
*	*	O	O	O	*	*
*	*	O	O	O	*	*
*	*	O	O	O	*	*
*	*	*	O	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*

Erosion for a Non-binary Image

$$A[m,n] = \min\left\{A[m,n],\, A[m-1,n],\, A[m+1,n],\, A[m,n-1],\, A[m,n+1]\right\}$$

附錄十七　常用的影像修飾方法

(2-2) Dilation　(擴大區域)

$$A[m,n] = A[m,n] \,||\, A[m-1,n] \,||\, A[m+1,n] \,||\, A[m,n-1] \,||\, A[m,n+1]$$

*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	0	*	*	*
*	*	*	0	0	0	0	*
*	*	0	0	0	0	*	*
*	*	*	*	0	0	0	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*

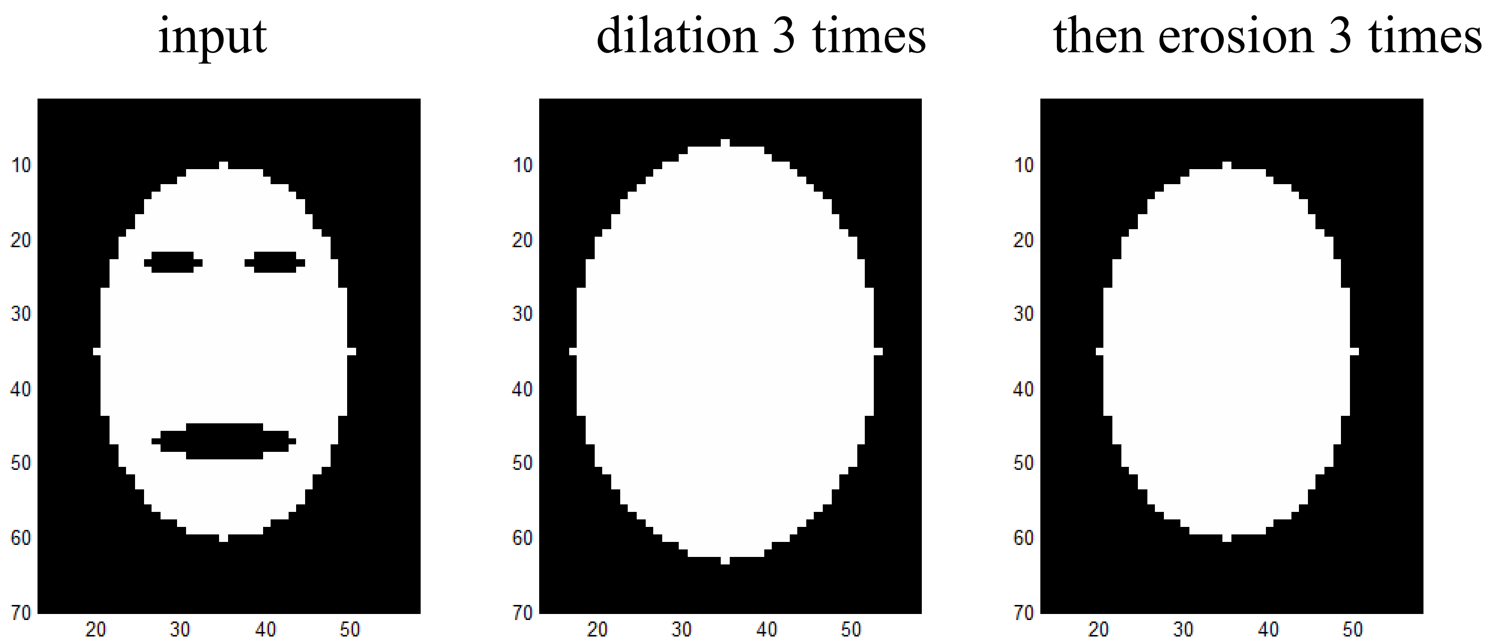
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	0	*	*	*
*	*	*	0	0	0	0	*
*	*	0	0	0	0	0	0
*	0	0	0	0	0	0	*
*	*	0	0	0	0	0	0
*	*	*	*	0	0	0	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*

Dilation for a Non-binary Image

$$A[m,n] = Max\{A[m,n],\, A[m-1,n],\, A[m+1,n],\, A[m,n-1],\, A[m,n+1]\}$$

(2-3) Closing (Hole Filling)

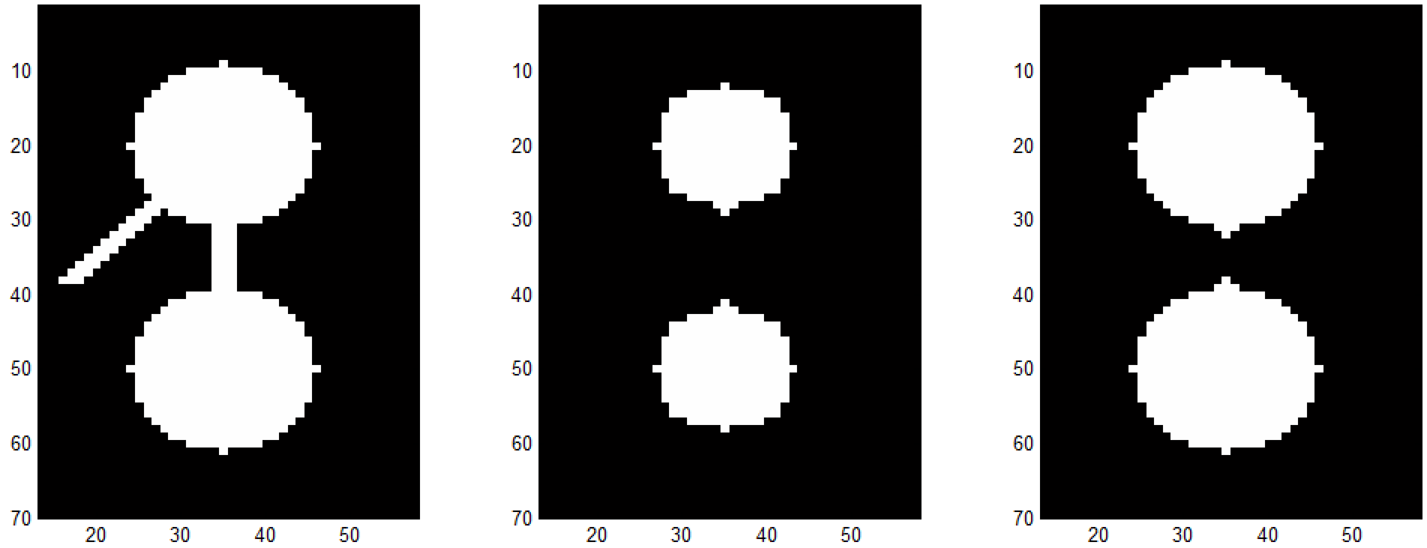
closing = dilation *k* times + erosion k times



(2-4) Opening

opening = erosion k times + dilation k times

input　　　　erosion 3 times　　　　then dilation 3 times



(3) Edge enhancement

$$input\ \ image + \alpha\left|edge\ \ detection\ \ output\right|$$

Original image

With edge enhancement

附錄十七　常用的影像修飾方法

(4) Dehaze (除霧)

He, Kaiming, Jian Sun, and Xiaoou Tang. "Single image haze removal using dark channel prior." *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 33, pp. 2341-2353 , 2011.

附錄十七　常用的影像修飾方法

Haze Model $\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(1 - t(\mathbf{x}))$

J(**x**): scene, **I**(**x**): observed image

t(**x**): transmission, **A**: intensity for the whole-haze case
A(1- **t**(**x**)): airlight

定義 dark channel $\mathbf{J}^{dark}(\mathbf{x})$

$$J^{dark}(\mathbf{x}) = \min_{c \in \{r,g,b\}} \left(\min_{\mathbf{y} \in \Omega(\mathbf{x})} (J^c(\mathbf{y})) \right),$$

$\Omega(\mathbf{x})$: some patch (a small region)

Dark channel 為一個影像在一個小範圍區域當中，RGB 的最小值

一個正常影像的 dark channel 大多近於 0
一個受 haze 影響的影像，dark channel 常常不為 0

附錄十七　常用的影像修飾方法

Dehaze 的方法與流程

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(1 - t(\mathbf{x}))$$

$$J^{dark}(\mathbf{x}) = \min_{c}\left(\min_{\mathbf{y}\in\Omega(\mathbf{x})} (J^c(\mathbf{y}))\right) = 0.$$

$$\min_{c}\left(\min_{\mathbf{y}\in\Omega(\mathbf{x})} \left(\frac{J^c(\mathbf{y})}{A^c}\right)\right) = 0$$

$$\frac{\mathbf{I}(\mathbf{x})}{A^c} = \frac{\mathbf{J}(\mathbf{x})}{A^c} t(\mathbf{x}) + 1 - t(\mathbf{x})$$

$$\boxed{\tilde{t}(\mathbf{x}) = 1 - \min_{c}\left(\min_{\mathbf{y}\in\Omega(\mathbf{x})} \left(\frac{I^c(\mathbf{y})}{A^c}\right)\right)}$$

find the transmission $t(\mathbf{x})$

A: the 95% largest intensity of $\mathbf{I}(\mathbf{x})$

$$\boxed{\mathbf{J}(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x})}{t(\mathbf{x})} + \mathbf{A}\left(1 - \frac{1}{t(\mathbf{x})}\right)}$$

recover the original image

He, Kaiming, Jian Sun, and Xiaoou Tang. "Single image haze removal using dark channel prior." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 2341-2353 , 2011.

期末的勉勵

- 人生難免會有挫折，最重要的是，**我們面對挫折的態度是什麼**

- 長遠的願景要美麗，短期的目標要務實

祝各位同學暑假愉快！

各位同學在研究上或工作上，有任何和 digital signal processing 或 time frequency analysis 方面的問題，歡迎找我來一起討論。