Walsh Haddamard Transform:

The Haddamard transform matrices H_n are $N \times N$ matrices

where $N = 2^n$, $n = 1, 2, 3, \ldots$

These can be generated from a core matrix

$$H_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

and $H_n = H_{n-1} \otimes H_1 = H_1 \otimes H_{n-1}$

eg: $H_2 = H_1 \otimes H_1 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$

$$H_3 = H_2 \otimes H_1 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & +1 & -1 & +1 \\ 1 & 1 & -1 & -1 & -1 & -1 & +1 & +1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

$$\underline{v} = H \underline{u}$$

$$H^{-1} = H^{H} = (H^{*})^{T}$$

here H is real $\ni$ $H^{-1} = H^{T}$

→ This transform doesn't have the energy packing capacity as that of DCT and it is generally not preferred.

→ But it is used for implementing hardware.

→ For 2D $\underline{v} = \mathcal{H} \underline{u}$ $\quad \mathcal{H} = H \otimes H$

Discrete cosine transform (DCT)

1D-DCT: used in jpeg, mpeg-2, H.263, ...

This DCT revolves around the two main properties,

1) Ability to concentrate energy in the first few coefficients.
2) Data decorrelation for strongly correlated Markov-1 process

It is of
$$\begin{bmatrix} 1 & \rho & \rho^2 & \rho & \cdots \\ \rho & 1 & \rho & \rho^2 & \cdots \\ \rho^2 & \rho & 1 & \rho & \\ \vdots & \vdots & & & \cdots \end{bmatrix}$$

$0 < \rho < 1$

* There are several forms of DCTs and we are dealing with DCT-II

1D-DCT is as follows:

$$V(K) = \alpha(K) \sum_{n=0}^{N-1} u(n) \cos\left[\frac{\pi(2n+1)K}{2N}\right] \quad ; \; 0 \le K \le N-1$$

where $\alpha(0) = \frac{1}{\sqrt{N}}$, $\alpha(K) = \sqrt{\frac{2}{N}}$, $1 \le K \le N-1$

1D-IDCT: $$u(n) = \sum_{K=0}^{N-1} \alpha(K) V(K) \cos\left[\frac{\pi(2n+1)K}{2N}\right] \quad ; \; 0 \le n \le N-1$$

$$V(K) = \alpha(K) \sum_{n=0}^{N-1} u(n) \cos\left(\frac{\pi(2n+1)K}{2N}\right)$$

can be written as $\underline{V} = C\, \underline{u}$

where
$$C(K,n) = \begin{cases} \frac{1}{\sqrt{N}} & ; \; K=0, \; 0 \le n \le N-1 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{\pi(2n+1)K}{2N}\right] & ; \; 1 \le K \le N-1, \; 0 \le n \le N-1 \end{cases}$$

$$C = \begin{array}{c} \xrightarrow{\;n\;} \\ K\downarrow \begin{bmatrix} \frac{1}{\sqrt{N}} & \frac{1}{\sqrt{N}} & \cdots & \frac{1}{\sqrt{N}} \\ \sqrt{\frac{2}{N}} & \cdots & \cdots & \cdots \\ \vdots & & & \\ \vdots & \cdots & \cdots & \cdots \end{bmatrix} \end{array}$$

Say $N = 3 \Rightarrow$ $K: 0$ to 2, $n =$ 0 to 2

$$C = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \sqrt{\frac{2}{3}}\cos\frac{\pi}{6} & \sqrt{\frac{2}{3}}\cos\frac{3\pi}{6} & \sqrt{\frac{2}{3}}\cos\frac{5\pi}{6} \\ \sqrt{\frac{2}{3}}\cos\frac{2\pi}{6} & \sqrt{\frac{2}{3}}\cos\pi & \sqrt{\frac{2}{3}}\cos\frac{10\pi}{6} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & -\sqrt{\frac{2}{3}} & \frac{1}{\sqrt{6}} \end{bmatrix}$$

Eigen values of R are ≥ 0

Proof: $\underbrace{\underline{x}^H R \underline{x}}_{\text{PSD}} = \underline{x}^H \underline{y} \underline{y}^H \underline{x}$

let $\underline{z} = \underline{y}^H \underline{x} \Rightarrow \underline{z}^H = \underline{x}^H \underline{y}$

$= \underline{x}^H \quad \underline{z}^H \underline{z}$

Choose $\underline{x}$ as eigenvector of R

$\underline{x}^H R \underline{x} = \underline{x}^H \lambda \underline{x} \geq 0$

$\lambda \underbrace{\underline{x}^H \underline{x}}_{\neq 0} \geq 0$

$\Rightarrow \lambda \geq 0$ //

---

6) Eigen filter for detect defection.

Singular Value Decomposition (SVD):

There are different interpretations of SVD:

1) The SVD is a generalization of the notion of diagonalization to rectangular matrices.

2) It can be looked as an data dependent Unitary transform [i.e the basis images are drawn from image itself]

3) It yields the best low-rank approximation of a matrix in the Frobenius norm sense.

If we have a rectangular matrix $S_{m \times n}$ can be written as

$$S_{m \times n} = A \Sigma B^H$$

A is $m \times m$, Σ is $m \times n$, B is $n \times n$

A & B are individually unitary i.e $A A^H = A^H A = I_{m \times m}$

$B^H B = B B^H = I_{n \times n}$

Σ is a diagonal matrix & the non-zero entries of Σ are called the singular values of $S_{m \times n}$.

$$SS^H = A\Sigma \underbrace{B^H B}_{I} \Sigma^H A^H$$

$$= \underbrace{A \Sigma \Sigma^H A^H}$$

eigen value & eigen vector decomp of S

[eigen vectors of SS^H are columns of A & eigen values of SS^H are in $\Sigma\Sigma^H$

$$S^H S = B\Sigma^H A^H A \Sigma B^H$$

$$= B\Sigma^H \Sigma B^H$$

[eigen vectors of $S^H S$ are col's of B and eigen values of $S^H S$ are in $\Sigma^H \Sigma$

eg: let S be 2x3 matrix

$$\Rightarrow \Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \end{bmatrix}_{2\times 3} \qquad \Sigma^H = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{bmatrix}$$

σ_1, σ_2 are called singular values of S

$$\Sigma\Sigma^H = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

σ_1^2, σ_2^2 are eigen values of SS^H

$$\Sigma^H\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$\sigma_1^2, \sigma_2^2, 0$ are eigen values of $S^H S$

→ Prove $S_{m\times n} = A\Sigma B^H = \sum_{i=1}^{p} \sigma_i \underline{a_i} \underline{b_i}^H$

where p is the rank of S

$\underline{a_i}$ is the i^{th} column of A

$\underline{b_i}$ is the i^{th} column of B

$(\underline{a_i} \underline{b_i}^H)$ is the basis images (which we got from A &

also $\langle S_{m\times n}, \underline{a_j} \underline{b_j}^H \rangle = \sigma_j$

→ If we consider the image transmission pblm, say we have an image $S_{64 \times 128}$, to transmit this the memory required will be

$\underline{a}_i$'s will 64×1, $\underline{b}_i$'s will 128×1, σ_i's require 4 bytes each if it is float,

$\underline{a}_i$'s − 64×4 bytes
$\underline{b}_i$'s − 128×4 bytes
σ_i's − 4 bytes

$\times P$

Hence SVD is not used in data compression because it needs lot of memory.

Consider $\hat{S}_{m \times n} = \sum_{i=1}^{r} \sigma_i \underline{a}_i \underline{b}_i^H$ where $r < P$

Error matrix $E = S - \hat{S} = \sum_{i=r+1}^{P} \sigma_i \underline{a}_i \underline{b}_i^H$

$$E_{m,n} = \sum_{i=r+1}^{P} \sigma_i a_{im} b_{in}^*$$

↓
$(m,n)^{th}$ entry of E

$$E_{m,n}^* = \sum_{i=r+1}^{P} \sigma_i a_{im}^* b_{in}$$

$$|E_{m,n}|^2 = E_{m,n} \cdot E_{m,n}^* = \left(\sum_{i=r+1}^{P} \sigma_i a_{im} b_{in}^*\right)\left(\sum_{j=r+1}^{P} \sigma_j a_{jm}^* b_{jn}\right)$$

$$= \sum_{i=r+1}^{P} \sigma_i^2 a_{im} a_{im}^* b_{in}^* b_{in} + \sum_{i=r+1}^{P} \sum_{\substack{j=r+1 \\ i \neq j}}^{P} \sigma_i \sigma_j a_{im} a_{jm}^* b_{in}^* b_{jn}$$

$$\cancel{= \sum_{i=r+1}^{P} \sigma_i^2 |a_{m}|^2 + \sum}$$

$$= \sum_{i=r+1}^{P} \sigma_i^2 |a_{im}|^2 |b_{in}|^2 + \sum_{i=r+1}^{P} \sum_{\substack{j=r+1 \\ i \neq j}}^{P} \sigma_i \sigma_j a_{im} a_{jm}^* b_{in}^* b_{jn}$$

$$\sum_m \sum_n |E_{m,n}|^2 = \sum_m \sum_n \left[\sum_{i=r+1}^{p} \sigma_i^2 |a_{im}|^2 |b_{in}|^2 + \sum_{\substack{i=r+1 \\ i \neq j}}^{p} \sum_{j=r+1}^{p} \sigma_i \sigma_j a_{im} a_{jm}^* b_{in} b_{jn}^* \right.$$

$$= \sum_{i=r+1}^{p} \sigma_i^2 \underbrace{\sum_m |a_{im}|^2}_{1} \underbrace{\sum_n |b_{in}|^2}_{1} + \sum_{\substack{i \\ i \neq j}} \sigma_i \sum_j \sigma_j \Big(\underbrace{\sum_m a_{im} a_{jm}^*}_{0} \Big) \sum_n \Big(\underbrace{b_{in} b_{jn}^*}_{0} \Big)$$

($\because$ $\underline{a_i}$, $\underline{b_i}$ are col's of A & B & which are unitary matrices)

$$= \sum_{i=r+1}^{p} \sigma_i^2$$

If we can find $\hat{S}$ such that $\| S - \hat{S} \|_F^2$ is minimum then that is the best approximation for S.

Applications of SVD:

1) Homography estimation (saw in initial lectures of this course)
2) It is used to compute eigen vectors while doing PCA.
3) Image expansion (~~seto~~ SVD is seldom used for this)

$$S_{m \times n} = A_{m \times m} \Sigma_{m \times n} B^H_{n \times n} = \sum_{i=1}^{r} \sigma_i \underline{a_i} \underline{b_i}^H$$

[r is rank of $S_{m \times n}$

suppose if we assume ~~as~~ S as data matrix

like for eg $S = \begin{bmatrix} | & | & - - & | \end{bmatrix}$ $\quad \underline{x_i}$ $\quad 4096 \times P$

S contains several face images, each image is stacked as a column of S

$P >> 4096$

face 1: 64×64

$\Rightarrow$ each col $\Rightarrow$ ~~4096~~ $\times 1$

Suppose we generate a new matrix $\underline{S}$ such that $\underline{S} = S - \underline{\mu}$

where $\underline{\mu}$ is the average of all the columns in S.

then covariance of the data can be given by $R = \underline{S}\, \underline{S}^H$

or in other words

$$R = \frac{1}{P} \sum (\underline{x_i} - \underline{\mu})(\underline{x_i} - \underline{\mu})^H$$

$R = \underline{S}\,\underline{S}^H$ | if we do SVD then $\underline{S} = A \Sigma B^H$

$\underline{S}^H = B \Sigma^H A$

$$R = A \Sigma \underbrace{B^H B}_{I} \Sigma^H A$$

$$= A \underbrace{\Sigma \Sigma^H}_{4096 \times 4096} A$$

$\Rightarrow R$ will also be 4096×4096

So to find eigen vectors we can use this SVD

5) Denoising

6) pseudo inverse (also called as Moore-penrose pseudo inverse)

If we have a matrix $S_{m \times n}$ expressed using SVD as

$$S_{m \times n} = \underset{m \times m}{A} \underset{m \times n}{\Sigma} B^H_{n \times n}$$

Pseudo inverse is represented as $S^+_{n \times m} = B \Sigma^+_{n \times m} A^H$

say $S_{2 \times 3}$ matrix then $\underset{2 \times 3}{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \end{bmatrix}$

$$\underset{3 \times 2}{\Sigma^+} = \begin{bmatrix} 1/\sigma_1 & 0 \\ 0 & 1/\sigma_2 \\ 0 & 0 \end{bmatrix}_{3 \times 2}$$

→ S^+ is weak inverse

→ It satisfies $S S^+ S = S$ & $S^+ S S^+ = S^+$

→ If true inverse exists (when $m = n$) then pseudo inverse will be equal to true inverse.