Nptel Important notes :-

→ image representation , $f(x,y) = r(x,y) * i(x,y)$

r = reflectivity $\qquad$ i = intensity

→ $I_{min} \le f(x,y) \le I_{max}$ → I = intensity

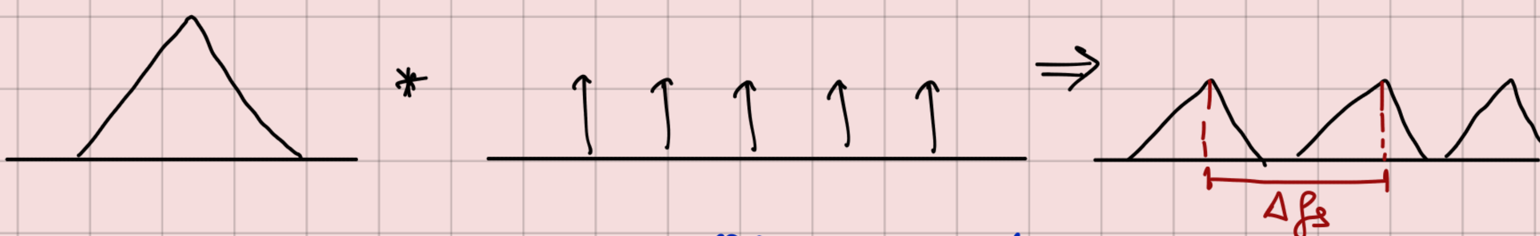
→ Sampler → Quantization → Digital computer → DAC converter → Display

→ sampling : converting analog input te discrete samples

$(f_s > 2f_m)$ → for best reconstruction

here ,

we multiply input signal te a train of impulse with frequency $f_s > 2f_m$ te get input signal at each impulse position



→ fourier transform, $X(\omega) = \int_{-\infty}^{\infty} x(t)\, e^{-j\omega t}\, dt$

→ fourier series , $v(t) = \sum_{n=-\infty}^{\infty} c(n)\, e^{jn\omega_0 t}$

$$c(n) = \frac{1}{T_0} \int_{T_0} v(t)\, e^{-jn\omega_0 t}\, dt$$

→ DFT pair :- $X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\left(\frac{2\pi}{N}\right)nk}$

$$x(n) = \sum_{k=0}^{N-1} X(k)\, e^{j\left(\frac{2\pi}{N}\right)nk}$$

→ convolution :- $y(n) = x(n) * h(n)$

$$\therefore\ y(n) = \int_{-\infty}^{\infty} x(z) \cdot h(t-z)\, dz$$

$$\{ x(n) * h(n) \longleftrightarrow X(\omega) \cdot H(\omega) \}$$

→ aliasing is an irreversible error occurs when $(f_s < 2f_m)$.
hence reconstruction of i/p is impossible.

→ for image $f_s(x,y) = \sum_{m,n=-\infty}^{\infty} \sum f(m\Delta x, n\Delta y)\, \delta(x - m\Delta x, y - n\Delta y)$

$$\{ (\omega_{xs} > 2\omega_{x0}) \text{ and } (\omega_{ys} > 2\omega_{y0}) \}$$

↳ Nyquist rate

→ we can recover i/p using a low pass filter,

$$H(\omega_x, \omega_y) = \begin{cases} \dfrac{1}{\omega_{xs} \cdot \omega_{xy}} & , (\omega_x, \omega_y) \in R \\ 0 & , \text{else} \end{cases}$$

→ DPI = Dots per inch

→ for quantization, $u' = r_k$ if $t_k < u < t_{k+1}$

→ It follows staircase representation

→ mean square error, $\left(\varepsilon_q = E[(u-u')^2] = \int_{t_1}^{t_{L+1}} (u-u')^2 p_u(u)\, du \right)$.

Now, $\left(\varepsilon_q = \int_{i}^{i+1} (u - r_i)^2 p_u(u)\, du \right)$

→ $t_k = \dfrac{r_k + r_{k-1}}{2}$, $\left(r_k = \dfrac{\int_{t_k}^{t_{k+1}} u\, p_u(u)\, du}{\int_{t_k}^{t_{k+1}} p_u(u)\, du} \right)$

→ Quantizer mean square distortion is given as

$$\xi_q = \frac{1}{12L^2} \left\{ \int_{t_1}^{t_{L+1}} [p_u(u)]^{1/3} du \right\}^3$$

→ Gaussian :- $p_u(u) = \dfrac{1}{\sqrt{2\pi}\,\sigma} e^{-(u-\mu)^2/2\sigma^2}$

→ Laplacian :- $p_u(u) = \dfrac{\alpha}{2} e^{(-\alpha |u-\mu|)}$

here, $\left(\sigma^2 = \dfrac{\alpha}{2}\right)$

→ uniform :- $p_u(u) = \begin{cases} \dfrac{1}{t_{L+1} - t_1} & , \ t_1 \le u \le t_{L+1} \\ 0 & , \ \text{else} \end{cases}$

then, $\left(r_k = \dfrac{t_{k+1} + t_k}{2} \right)$, $\left(t_k = \dfrac{r_{k+1} + r_k}{2} \right)$, $\left(t_k = \dfrac{t_{k+1} + t_{k-1}}{2} \right)$

$\left(q = \dfrac{t_{L+1} - t_1}{L} \right)$, $\left(t_k = (k+1)q \right)$, $\left(r_k = t_k + \dfrac{q}{2} \right)$

$\left(t_k = t_{k-1} + q \right)$ $\forall\ k = 1, 2, \ldots, L+1$

→ Quantization error of uniform is from $-\dfrac{q}{2}$ to $\dfrac{q}{2}$

$\therefore \left(\xi_q = \dfrac{q^2}{12} \right)$ also $SNR = 10 \log_{10} 2^{2B} = \dfrac{P_S}{P_N}$

$= 6B$ dB

→

	$(x-1, y)$	
$(x, y-1)$	P	$(x, y+1)$
	$(x+1, y)$	

$N_4(P)$

$(x-1, y-1)$		$(x-1, y+1)$
	P	
$(x+1, y-1)$		$(x+1, y+1)$

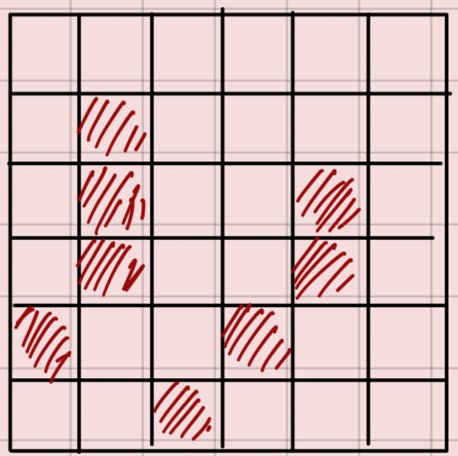
$N_D(P)$

→ $N_8(P) = N_4(P) \cup N_D(P)$

→ when two pixels are adjacent in some sense, then it is connected. How?

(i) either $N_4(P)$, $N_8(P)$, $N_D(P)$

(ii) intensity is same.



→

	1				
	1			2	
	1			2	
3			4		
		5			

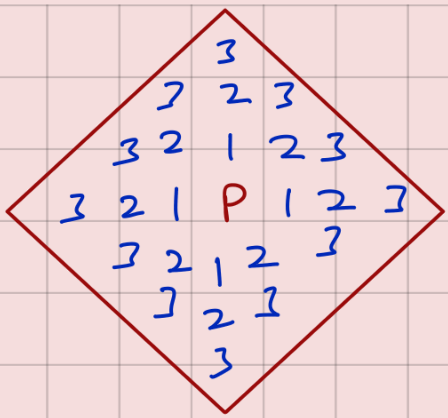
	1				
	1			2	
	1			2	
1			2		
		2			

→ Now we see (1, 3) and (2, 4, 5) are connected hence we club them.

→ Distance measures :-

(i) Euclidean → $D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$

(ii) City block → $D_n(p,q) = |x-s| + |y-t|$

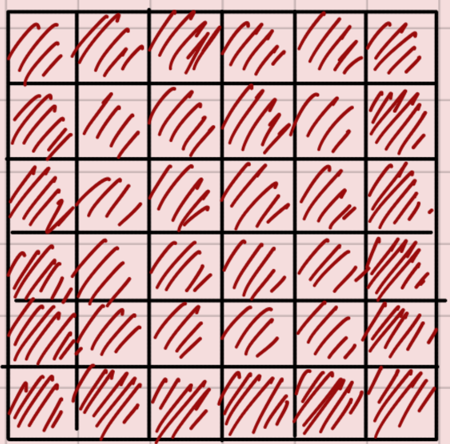


(iii) Chess Board :- $\max(|x-s|, |y-t|)$

2	2	2	2	2
2	1	1	1	2
2	1	P	1	2
2	1	1	1	2
2	2	2	2	2

City block > Euclidean > chess

Distance transformation



$\longrightarrow$

1	1	1	1	1	1
1	2	2	2	2	1
1	2	3	3	2	1
1	2	3	3	2	1
1	2	2	2	2	1
1	1	1	1	1	1

Operators

$\overbrace{p+q,\ p*q,\ p\%q,\ p-q}^{\text{Arithmetic}},\ \overbrace{p\cdot q,\ p',\ p+q}^{\text{Logical}}$

translation matrix :-

$x^* = x + x_0, \quad y^* = y + y_0, \quad z^* = z + z_0$

$$\begin{pmatrix} x^* \\ y^* \\ z^* \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Scaling matrix :-

$$S = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation matrix :-

$$R_z = \begin{pmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad R_y = \begin{pmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

→ for inverse transform $x \to -x$ and $\theta \to -\theta$

→ concatenation of these operation is not commutative in nature.

Eg → $R(S(T)) \neq S(T(R))$

→ consider x, y, z be camera coordinate system then perspective transformation is given,

$$x = \frac{X\lambda}{\lambda - Z}, \quad y = \frac{Y\lambda}{\lambda - Z}$$

where X, Y, Z are world coordinate system.

and λ = focal length.

→ homogeneous coordinate → (kX, kY, kZ, k)

→ $X = \frac{x_0}{\lambda}(\lambda - Z)$, $Y = \frac{y_0}{\lambda}(\lambda - Z)$,

→ for a given point 'w' in world coordinate, to find the corresponding image point c, we do

$$c_h = P(T(R(G(w_h))))$$

G = inverse translation matrix to $[-x_0, -y_0, -z_0] \to w_h$

$R = R_\alpha R_\theta$, R_θ = rotation in z axis

R_α = rotation in x-axis

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/\lambda & 1 \end{pmatrix}$$

T = inverse translation matrix to $[-r_1, -r_2, -r_3]$

P = perspective transformation matrix

→ for stereo image setup, $X_1 = \frac{x_1}{\lambda}(\lambda - z)$, $X_2 = \frac{x_2}{\lambda}(\lambda - z)$

$$\left(z = \lambda - \frac{\lambda B}{x_2 - x_1} \right) , \quad x_2 - x_1 = \text{disparity}$$

$$\left(X = \frac{x_1}{\lambda}(\lambda - z) \right) , \quad \left(Y = \frac{y_1}{\lambda}(\lambda - z) \right)$$

→ Now here, λ = focal length, B = displacement of camera.

→ Interpolation refers to the process of estimating pixel values between two integral pixel values.

→ spline function :- $x(t) = \sum_{i=0}^{n} P_i \, B_{i,k}(t)$

constant

↳ normalized B-spline of order k.

At, k = 1

$$B_{i,1} = \begin{cases} 1 & , \; t_i \le t < 1 \\ 0 & , \; \text{else} \end{cases}$$

$\therefore \boxed{B_{ik}(t) = B_{0,k}(t - i)}$

fixing i = 0 ⤵

(i) k = 0

$$B_{0,1}(t) = \begin{cases} 1 & , \; 0 \le t < 1 \\ 0 & , \; \text{else} \end{cases}$$

(ii) k = 1

$$B_{0,2}(t) = \begin{cases} t & , \; 0 \le t < 1 \\ 2 - t & , \; 1 \le t < 2 \\ 0 & , \; \text{else} \end{cases}$$

(iii) $k=3$

$$B_{0,3}(t) = \begin{cases} t^2/2 & , \quad 0 \le t < 1 \\ \dfrac{-2t^2+6t-3}{2} & , \quad 1 \le t < 2 \\ \dfrac{(3-t)^2}{2} & , \quad 2 \le t < 3 \\ 0 & , \quad \text{else} \end{cases}$$

(iv) $k=4$

$$B_{0,4}(t) = \begin{cases} t^3/6 & , \quad 0 \le t < 1 \\ \dfrac{-3t^3+12t^2-12t+4}{6} & , \quad 1 \le t < 2 \\ \dfrac{3t^3-24t^2+60t-44}{6} & , \quad 2 \le t < 3 \\ \dfrac{(4-t)^3}{6} & , \quad 3 \le t < 4 \\ 0 & , \quad \text{else} \end{cases}$$

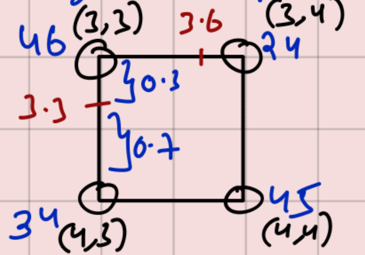
→ constant B-Spline → 2 points
→ linear B-spline → 3 points
→ quadratic B-spline → 4 points → only unsymmetric one.
→ Cube B-spline → 5 points

→ $x(t) = \sum_{i=0}^{n} P_i \, B_{i,k}(t) = \sum_{i=0}^{n} P_i, \, B_{i-s,k}(t)$

here, if
$$\left.\begin{array}{l} k=1, \; s=0.5 \\ k=2, \; s=1 \\ k=4, \; s=2 \end{array}\right\}$$ only works for symmetric one

→ for image interpolation let us see how to perform.

Eg →



and we are asked to find for $(3.3, 3.6)$

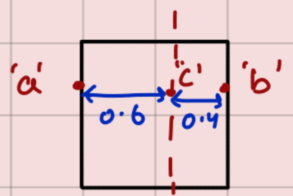
(i) find point b/w 46 and 34 as,

$$a = 46 \times 0.7 + 34 \times 0.3 = 42.4$$

(ii) find point b/w 24 and 45 as,

$$b = 24 \times 0.7 + 45 \times 0.3 = 30.3$$

(iii) Now using point 'a' and 'b' find find 'c' as,



$$\therefore \left(c = 42.4 \times 0.4 + 30.3 \times 0.6 = \underline{35.14} \right)$$

final pixel point value.

→ Image transformation represents a given image as a series summation of a set of unitary matrix

→ a matrix is unitary iff, $(A^{-1} = A^{*T})$

→ two function is orthogonal iff $\int_T a_m(t) \cdot a_n(t)\, dt = \begin{cases} k & , m = n \\ 0 & , m \neq n \end{cases}$

→ if $u(m,n)$ is an unitary matrix then

$$v(k,l) = \sum_{m,n=0}^{N-1} \sum a_{k,l}(m,n)\, u(m,n), \quad \forall\ 0 \le k, l, \le N-1$$

transformed unitary

transformation unitary matrix

and

$$u(m,n) = \sum_{k,l=0}^{N-1} \sum a^*_{k,l}(m,n)\, v(k,l) \quad , \forall\ 0 \le m, n, \le N-1$$

This takes N^4 computations the complete

→ seperable unitary transformation takes $(2N^3)$ computations.

here, $V = A(U A^T)$ or $(A(AU)^T)^T$

and $U = A^{*T} V A^*$

→ fourier transform pair is given as :-

$$F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-j2\pi ux}\, dx \quad \Bigg| \quad f(x) = \int_{-\infty}^{\infty} F(u)\, e^{j2\pi ux}\, du$$

→ mag of $F(u) = \sqrt{Real^2(u) + Img^2(u)}$

→ $\phi = \tan^{-1}\left(\frac{Img(u)}{Real(u)}\right)$, → power spectrum, $P(u) = Real^2(u) + Img^2(u)$

→ similarly for 2-D image :-

$$F(u,v) = \iint_{-\infty}^{\infty} f(x,y)\, e^{-j2\pi(ux+vy)}\, dx\, dy \quad \Big| \quad f(x,y) = \iint_{-\infty}^{\infty} F(u,v)\, e^{j2\pi(ux+vy)}\, du\, dv$$

→ other formula are valid here of 1-D.

→ for 2-D discrete Domain we have :-

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

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

→ properties :-

(i) $F(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} F(x,v)\, e^{-j\frac{2\pi ux}{N}}$

(ii) $f(x,y) = \frac{1}{N} \sum_{u=0}^{N-1} f(u,y)\, e^{j\frac{2\pi ux}{N}}$

(iii) $f(x-x_0, y-y_0) \longleftrightarrow F(u,v)\, e^{-j2\pi\left(\frac{ux_0}{M} + \frac{vy_0}{N}\right)}$

(iv) $F(u-u_0, v-v_0) \longleftrightarrow f(x,y)\, e^{j2\pi\left(\frac{u_0x}{M} + \frac{v_0y}{N}\right)}$

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

(vi) $f^*(x,y) \longleftrightarrow F^*(-u,-v)$

(vii) if we rotate $f(x,y)$ by 'θ_0' , then $F(u,v)$ also rotates by 'θ_0'

(viii) $f(ax, by) \longleftrightarrow \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right)$

(ix) $\bar{f}(x,y) \rightarrow$ average $= \frac{1}{MN} F(0,0)$

→ DFT takes total '$N \log_2 N$' computation, better than above two method.

→ Discrete cosine transformation :-

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cos\left(\frac{(2x+1)u\pi}{2M}\right) \cdot \cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

→ Hadamard matrix transformation

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{and} \quad H_{2N} = \begin{pmatrix} H_N & H_N \\ H_N & H_{-N} \end{pmatrix}$$

Eg → $H_4 = \begin{pmatrix} \boxed{\begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix}} & \boxed{\begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix}} \\ \boxed{\begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix}} & \boxed{\begin{matrix} -1 & -1 \\ -1 & 1 \end{matrix}} \end{pmatrix}$

→ modified hadamard matrix :-

here we first write the normal hadamard matrix and sort the rows in accordance to number of sign change.

Eg → $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{matrix} \rightarrow 0 \\ \rightarrow 3 \\ \rightarrow 1 \\ \rightarrow 2 \end{matrix}$ $\quad \therefore$ we get $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$

(normal hadamard) (modified hadamard)

→ Energy compaction of DCT is much higher than DFT, DWT and DHT

fourier

walsh

hadamard

→ for KL transform :-

Eg→ $\left\{ \begin{pmatrix}1\\6\end{pmatrix} \begin{pmatrix}2\\7\end{pmatrix} \begin{pmatrix}3\\8\end{pmatrix} \begin{pmatrix}4\\9\end{pmatrix} \begin{pmatrix}5\\10\end{pmatrix} \right\}$

$$a = \frac{1+2+3+4+5}{5} = 3, \quad b = \frac{6+7+8+9+10}{5} = 8$$

hence $\mu_x = \begin{pmatrix}a\\b\end{pmatrix} = \begin{pmatrix}3\\8\end{pmatrix}$

Now, $C_x = E\{(x-\mu_x)(x-\mu_x)^T\}$

let us find $(x-\mu_x)$,

→ $\begin{pmatrix}-2\\-2\end{pmatrix}, \begin{pmatrix}-1\\-1\end{pmatrix}, \begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}$

→ $\begin{pmatrix}-2\\-2\end{pmatrix}(-2 \ \ -2) = \begin{pmatrix}4 & 4\\4 & 4\end{pmatrix}, \begin{pmatrix}-1\\-1\end{pmatrix}(-1 \ \ -1) = \begin{pmatrix}1 & 1\\1 & 1\end{pmatrix}$

$\begin{pmatrix}1\\1\end{pmatrix}(1 \ \ 1) = \begin{pmatrix}1 & 1\\1 & 1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}(2 \ \ 2) = \begin{pmatrix}4 & 4\\4 & 4\end{pmatrix}$

$\begin{pmatrix}0\\0\end{pmatrix}(0 \ \ 0) = \begin{pmatrix}0 & 0\\0 & 0\end{pmatrix}$

Now, $C_x = \begin{pmatrix}2 & 2\\2 & 2\end{pmatrix}$, Now, $|C - \lambda I| = 0$

$$\left| \begin{pmatrix}2 & 2\\2 & 2\end{pmatrix} - \begin{pmatrix}\lambda & 0\\0 & \lambda\end{pmatrix} \right| = 0 \rightarrow \begin{vmatrix}2-\lambda & 2\\2 & 2-\lambda\end{vmatrix} = 0$$

$\therefore (2-\lambda)^2 = 4 \rightarrow (2-\lambda) = \pm 2$

$\therefore \boxed{\lambda = 4, 0}$

(i) $\lambda = 4$

$CA = \lambda A \rightarrow \begin{pmatrix}2 & 2\\2 & 2\end{pmatrix}\begin{pmatrix}a\\b\end{pmatrix} = 4\begin{pmatrix}a\\b\end{pmatrix}$

$\rightarrow \begin{pmatrix}2a+2b\\2a+2b\end{pmatrix} = \begin{pmatrix}4a\\4b\end{pmatrix} \rightarrow \therefore (a = b)$

let $a=1, b=1$

(ii) $\lambda = 0$

$2a+2b=0 \rightarrow (a=-b)$ let $a=1,\ b=-1$

$$\therefore e_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad e_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\therefore A = (e_1\ \ e_2) = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Now, $Y_i = A(x_i - u_x)$

→ In KL the kernel is not fixed, but depend on value of x.

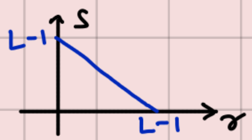
→ Energy compaction property is much higher than other.

→ computation complexity is much higher.

→ Consider mask of size 3X3 with values as $w_{i,j}$ then the processed image will be

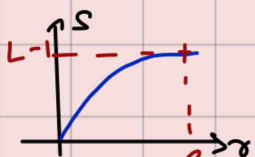
$$g(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} w_{i,j}\, f(x+i, y+j)$$

→ image negative : $s = T(r) = L-1-r$



→ "dynamic range" must be low in-order to see details / contrast

→ dynamic range compression : $s = T(r) = c \log(1+|r|)$



→ for gama correction / power law transformation, if $\gamma < 1$ then on the darker side, the lower range of intensity will be mapped into larger range of intensity values in the processed image and vice versa for lighter side.

→ also reverse is true for $\gamma > 1$.

→ normalized histogram, $P(r_k) = \dfrac{n_k}{n}$

→ $s_k = T(r_k) = \sum_{i=0}^{k} \dfrac{n_i}{n}$ ← target histogram.

→ $s' = \text{Int}\left[\dfrac{s - s_{min}}{1 - s_{min}} \times (L-1) + 0.5\right]$

6 : 15

Eg → $r = 0, 1, \ldots, 7$, $s = 0, 1, \ldots, 7$

$P_r(0) = 0$, $P_r(1) = P_r(2) = 0.1$, $P_r(3) = 0.3$, $P_r(4) = P_r(5) = 0$, $P_r(6) = 0.4$, $P_r(7) = 0.1$

r	$P_r(r)$	$T(r) = \sum_{i=0}^{r} P_r(i)$	$s' = \text{Int}\left[\dfrac{s - s_{min}}{1 - s_{min}} \times 7 + 0.5\right]$
0	0	0	0
1	0.1	0.1	1
2	0.1	0.2	2
3	0.3	0.5	4
4	0	0.5	4
5	0	0.5	4
6	0.4	0.9	7
7	0.1	1	7

→ averaging filter :-

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

→ weighted average :-

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

$$g(x, y) = \frac{\sum_{i=-a}^{a} \sum_{j=-b}^{b} w_{i,j}\, f(x+i, y+j)}{\sum_{i=-a}^{a} \sum_{j=-b}^{b} w_{i,j}}$$

→ median filter

$$\begin{bmatrix} 100 & 85 & 98 \\ 99 & 105 & 102 \\ 90 & 101 & 108 \end{bmatrix}_{3\times 3} \longrightarrow$$

median

85	90	98	99	100	101	102	105	108
1	2	3	4	5	6	7	8	9

9 × 1

→ $$\frac{df}{dx} = f(x+1) - f(x) \rightarrow 1^{st} \text{ order}$$

$$\frac{d^2f}{dx^2} = f(x+1) + f(x-1) - 2f(x)$$

→ $$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

→ $$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{-ve} \\ f(x,y) + \nabla^2 f(x,y) & \text{+ve} \end{cases}$$

(Laplacian)

→ Composite mask

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

→ unsharp masking $f_s(x,y) = \underbrace{f(x,y)}_{\text{original}} - \underbrace{\bar{f}(x,y)}_{\text{blurred}}$

→ high boost filter of unsharp :- $f_{hb}(x,y) = A f(x,y) - \bar{f}(x,y)$

→ high boost of Laplacian :-

$$f_{hb}(x,y) = \begin{cases} A f(x,y) - \nabla^2 f(x,y) \\ A f(x,y) + \nabla^2 f(x,y) \end{cases}$$

→ ∇f = gradient : $\left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$

→ $\partial f / \partial x$

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

$\partial f / \partial y$

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

shobel operator

→ Low pass filter

$$H(u) = A e^{-u^2/2\sigma^2}$$

$$h(x) = \sqrt{2\pi}\, A e^{-2\pi^2\sigma^2 x^2}$$

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

→ high pass filter

$$H(u) = A\left(1 - e^{-u^2/2\sigma^2}\right)$$

$$h(x) = A\left[\delta(x) - \sqrt{2\pi}\, A e^{-2\pi^2\sigma^2 x^2}\right]$$

0	1	0
1	-4	1
0	1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

→ distance of point u, v from origin

$$D(u,v) = \sqrt{\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2}$$

→ Butterworth lowpass filter

$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)}{D_0}\right]^{2n}}$$

order → n

→ Butterworth highpass

$$H(u,v) = \frac{1}{1 + \left(\frac{D_0}{D(u,v)}\right)^{2n}}$$

→ Gaussian lowpass

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

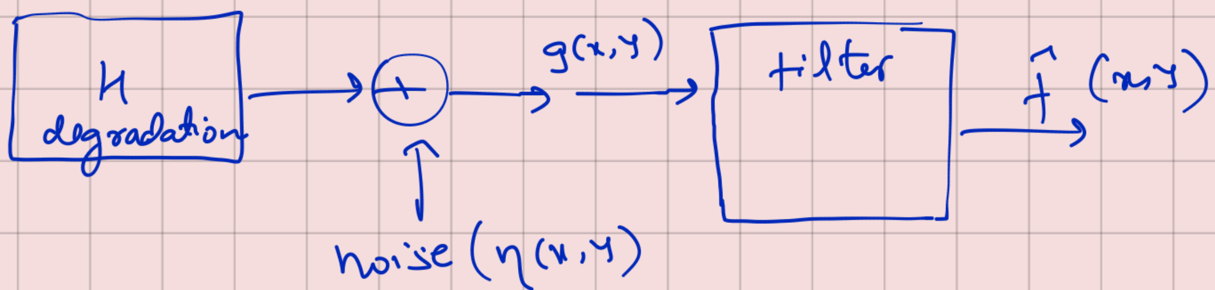
→ Gaussian highpass

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

→ for highboost filter :-

$$f_{hb}(x,y) = A\,f(x,y) - f_{lp}(x,y)$$
$$= (A-1)\,f(x,y) - f_{hp}(x,y)$$

→ image degradation, $g(x,y) = h(x,y) * F(x,y) + \eta(x,y)$



$$g(x,y) = \int\int_{-\infty}^{\infty} f(u,v) \cdot h(x-u, y-v)\, du\, dv + \eta(x,y)$$

→ discrete formulation

$$g_e(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e(m,n)\, h_e(x-m, y-n)$$

→ observation method :-

$$H_s(u,v) = \frac{G_s(u,v)}{F_s(u,v)}$$

→ Experimentation method :-

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

→ Mathematical modelling :-

$$H(u,v) = e^{-K(u^2+v^2)^{5/6}}$$

K → nature of turbulance

→ Basic principle method

$$H(u,v) = \frac{1}{\pi(ua+vb)} \sin[\pi(ua+vb)] e^{-j\pi(ua+vb)}$$

→ Motion blurr

$$H(u,v) = \frac{T}{\pi(ua+vb)} \sin(\pi(ua+vb)) e^{-j\pi(ua+vb)}$$

→ weiner filtering method :-

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \times \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right] \times G(u,v)$$

→ Image matching :-

we have three method to find image match

(i) $\max(|f-g|)$

(ii) $\iint_A |f-g|$

(iii) $\iint_A (f-g)^2$

→ Cauchy - Schwartz inequality :-

$$\iint_A f(x,y)\cdot g(x+u, y+v)\,dx\,dy \le \sqrt{\iint_A f^2(x,y)\,dx\cdot dy \cdot \iint_A g^2(x+u,y+v)\,dx\,dy}$$

let ↗ $= C_{fg}$ then normalized cross correlation is given.

$$\left[\frac{C_{fg}}{\sqrt{\iint_A g^2(x+u, y+v)\,dx\,dy}} \le \sqrt{\iint_A f^2(x,y)\,dx\,dy} \right] ✓$$

→ max value will give us the correct correlation

→ visible range → (400nm to 700nm) wavelength
→ radiance is the total energy coming out of light source (watt)
→ luminance is the total energy perceived by an observer (lumens)

→ RGB → primary color for light and secondary color for pigment
→ R + B → magenta, G + B = Cyan, R + G = Yellow
→ CMY → primary color for pigment and secondary color for light

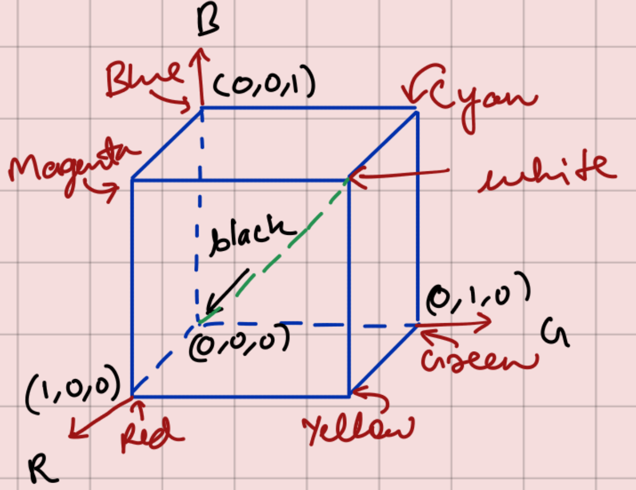
→ Brightness is the chromatic intensity
→ hue is the dominant wavelength in an image
→ Saturation gives the purity of color.

Chromatic coefficient :-

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z}$$

$$(x+y+z=1)$$

RGB Model :-



here we want to convert RGB to CMY,

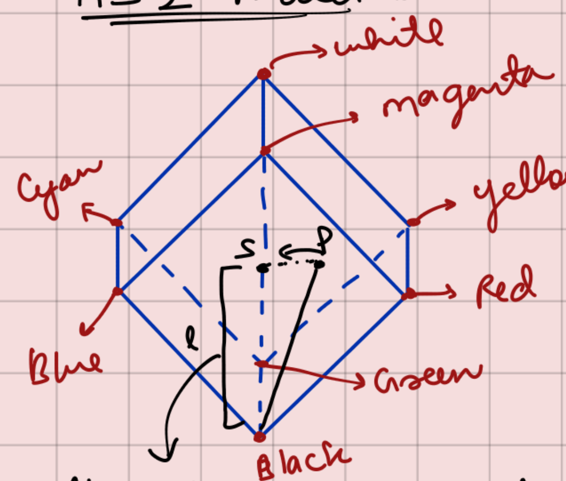
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

here both $\begin{bmatrix} C \\ M \\ Y \end{bmatrix}$ and $\begin{bmatrix} R \\ G \\ B \end{bmatrix}$ are in the normalized form.

→ The line joining black and white is intensity line. (green line)

→ equal amount of CMY should give black but in practical it gives muddy black.

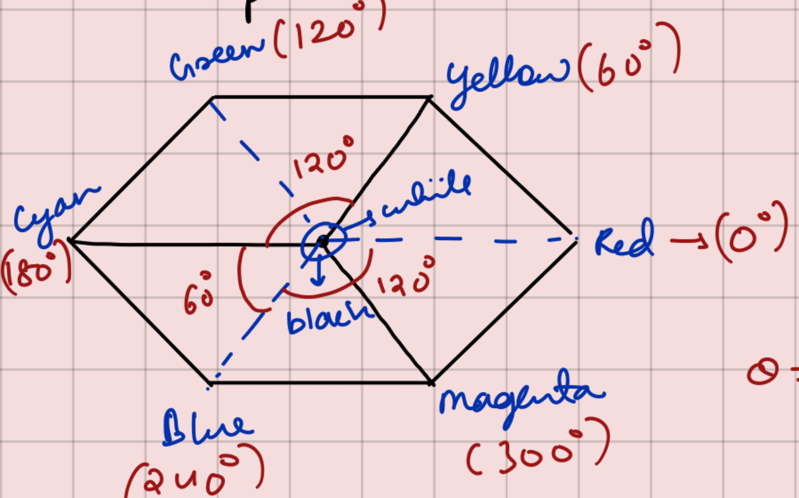
HSI model :-



length (l) = intensity of point 'P'

→ The line joining black and white is intensity line.

→ we made projection of 'P' on 'S'.



then,

$$H = \begin{cases} \theta^\circ & , \text{ if } B \le G \\ (360 - \theta)^\circ & , \text{ if } B > G \end{cases}$$

where,

$$\theta = \cos^{-1}\left\{ \frac{\frac{1}{2}\left[(R-G) + (R-B)\right]}{\left[(R-G)^2 + (R-B)(G-B)\right]^{1/2}} \right\}$$

and, $S = 1 - \dfrac{3}{(R+G+B)} \cdot [\min(R, G, B)]$

$I = \dfrac{1}{3}[R+G+B]$

Now, converting HSI to RGB.

(i) RG region $(0^\circ \le H \le 120^\circ)$

$$B = I(1-S), \quad G = 1-(R+B)$$

$$R = I\left(1 + \frac{S \cdot \cos(H)}{\cos(60^\circ - H)}\right)$$

(ii) GB region $(120^\circ \le H \le 240^\circ)$

$$H = H - 120^\circ, \quad G = I\left[1 + \frac{S \cdot \cos(H)}{\cos(60^\circ - H)}\right]$$

$$R = I(1-S)$$

$$B = 1-(R+G)$$

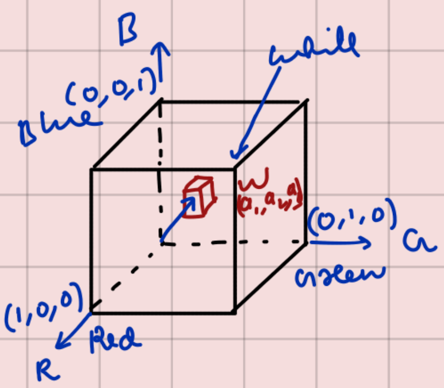
(iii) BR region $(240^\circ \le H \le 360^\circ)$

$$H = H - 240^\circ, \quad B = I\left[1 + \frac{S \cdot \cos(H)}{\cos(60^\circ - H)}\right]$$

$$G = I(1-S)$$

$$R = 1-(G+B)$$

<u>Color slicing</u> :-



$$s_i = \begin{cases} 0.5 & \text{if } \left[|r_j - a_j| > \frac{\omega}{2}\right], \ 1 \le j \le 3 \\ r_i & \text{otherwise.} \end{cases}$$

$\forall i = 1, 2, 3$

Tone correction :-

(i) flat tone , (ii) light tone, (iii) dark tone.

Smoothing :-

$$\bar{c}(x,y) = \frac{1}{k} \sum c(x,y)$$, $c(x,y) \rightarrow$ vector of RGB.

$$\forall (x,y) \in N_{x,y}$$

$$\bar{c}(x,y) = \begin{bmatrix} \frac{1}{k} \sum R(x,y) \\ \frac{1}{k} \sum G(x,y) \\ 1/k \sum B(x,y) \end{bmatrix} \forall (x,y) \in N_{xy}.$$