Lecture - 1 Overview

1. Signals
2. Digital Images
3. Processing digital images

* Analog to Digital conversion : Sampling and Quantization

Lecture - 2 Intensity Transformation

* Motivation

* Pointillism

→ Picture made by same intensity dots
→ First idea of digital image

1. Intensity transformation functions
2. Histogram processing

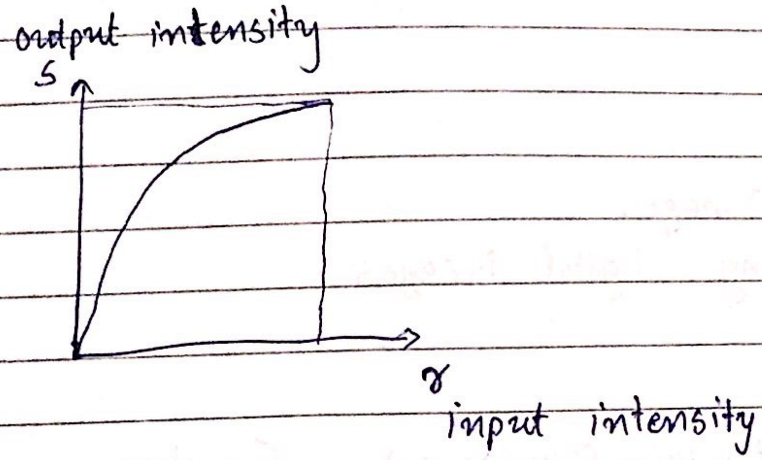
* Intensity Transformation function

→ input pixel (r) ⟶ output pixel (s)

→ a function T such that $T(r) = s$ is transformation function.

→ Transformation is independent of neighbouring pixels.

Ex.



⇒ Standard intensity transforms

* Image negatives → False image

* Log transforms

$$s = c \log(1+r)$$

* Power-Law (Gamma) Transformation - Gamma correction

$$s = c\, r^{\gamma}$$

* Piece-wise Transformation
  - Contrass Stretching

* Bit Plane slicing

  → Most significant bit represents underlined structure.
  → least significant bit represents details.

Scanned by CamScanner

* Probability density function

$p(x) \geq 0 \qquad \int_{-\infty}^{\infty} p(x) = 1$

* Commulative density function

$$C(x) = \int_{-\infty}^{x} p(x)\, dx$$

$$C(-\infty) = 0$$
$$C(\infty) = 1$$

* Histogram Processing

$h_r(i) = n_i$ 	i = intensity value $\in [0, L-1]$
n_i = # of pixels with intensity i

$H_r(i) = \sum_{j=0}^{i} h_r(i)$ 	(cummulative histogram)

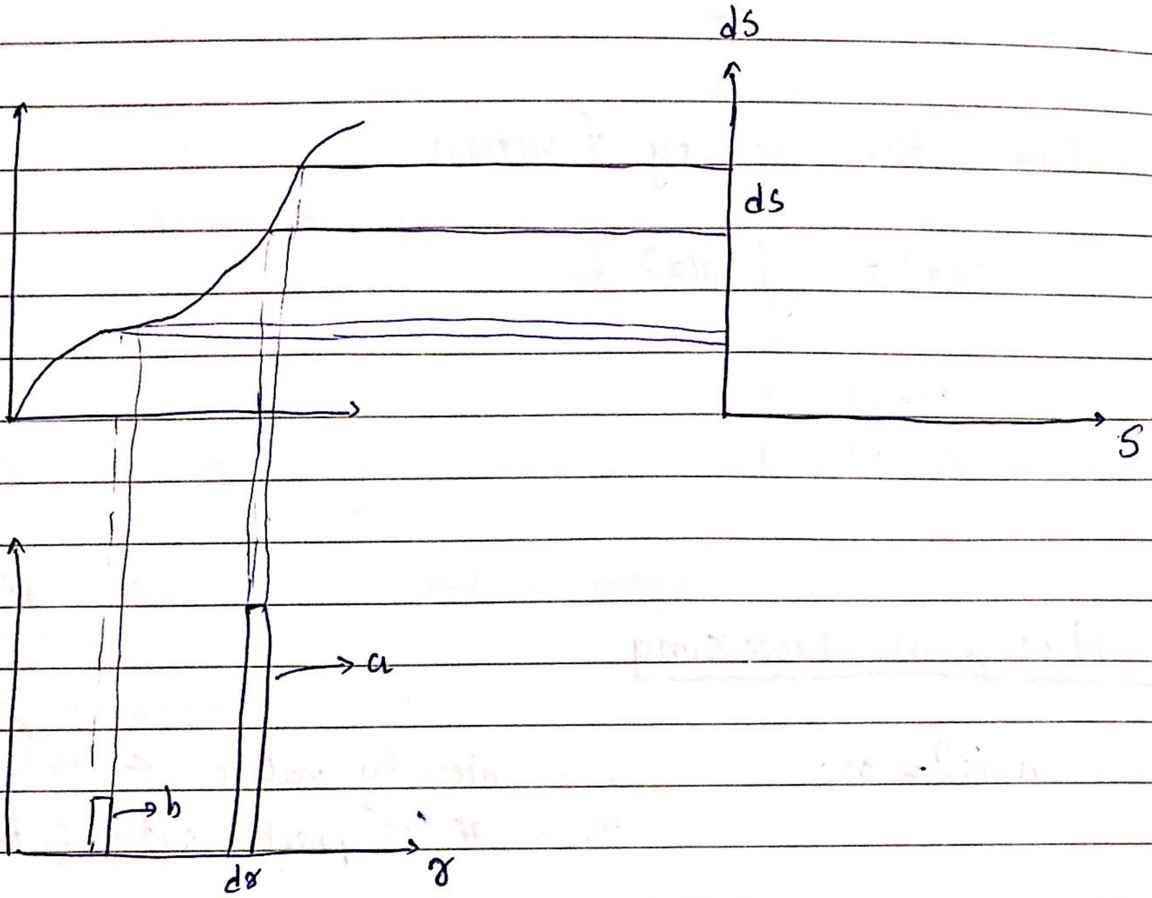
→ By dividing total # pixels, we get normalized histogram.

→ Spatial information is lost.

⇒ Histogram Equalization

(P.T.O.)

→ For pixel intensity range, whose histogram is high, need to be distributed along wider range of intensity.



→ So intensities with higher histogram (a) will have higher slope in commulative histogram, hence it will map to ~~high~~ wider range of output intensity.

Ex. a

→ Intensities with lower histogram (b) will have lower slope in commulative histogram, hence it will map to narrower range of output intensity.

Ex. b

$p(r)\, dr \approx p(s)\, ds$

* Histogram Matching

→ Keep relative shape same. (relative shape of comm. histogram)

* Histogram equalization for ~~some~~ small patches separately.

Lecture-3 Spatial Filtering

* Recap of Histogram Equalization

* Idea of neighbourhood
  → 4 neighbours
  → 8 neighbours

* Spatial Filtering mask (spatial masks)

* Gaussian filter

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$g(x,y) = \frac{1}{\sqrt{2\pi\sigma_x\sigma_y}} e^{\left(-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}\right)}$$

Ex: take example $\sigma_x = \sigma_y = 1$ $\mu_x = \mu_y = 0$

$$\frac{1}{\sqrt{2\pi}} \begin{pmatrix} e^{-1} & e^{-1/2} & e^{-1} \\ e^{-1/2} & e^{0} & e^{-1/2} \\ e^{-1} & e^{-1/2} & e^{-1} \end{pmatrix}$$

* Averaging filter (smoothing)

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

⇒ Bigger filter gives more smoothness.

→ Filters with large σ will give smoothness to image, als

⇒ Noise removal

→ How to sharpen ~~images~~ images ?

→ Edge sharpening

→ Edges are sudden intensity change. i.e. differentiatio is high.

Ex.
$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

$$\therefore \frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x,y)$$

$$\frac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y)$$

$$\frac{\partial^2 f(x,y)}{\partial x} = \left(\frac{\partial f(x+1,y)}{\partial x} - \frac{\partial f(x,y)}{\partial x}\right) - \left(\frac{\partial f(x,y)}{\partial x} - \frac{\partial f(x-1,y)}{\partial x}\right)$$

$$= f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\frac{\partial^2 f(x,y)}{\partial y} = \frac{\partial f(x,y+1)}{\partial y} + \frac{\partial f(x,y-1)}{\partial y} - \frac{2\partial f(x,y)}{\partial y}$$

* Laplacian filters

$$\nabla^2 f(x,y) = \frac{\partial f(x,y)}{\partial x^2} + \frac{\partial f(x,y)}{\partial y^2}$$

I_0 = original
I_s = sharpened

$$I_s = I_0 + c\,\nabla^2 I_0$$

$$= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

0	1	0
1	-4	1
0	1	0

* Unsharp Masking (Highboost filtering)

→ Blur original image.
→ ~~Sptr~~ Subtract original ~~img~~ image from blurred one. It will give edges of images.

* Robert Cross ~~Radient~~ Gradient
* Sobel ~~Radient~~ Gradient

* Other spatial filters (non-linear)

→ Max filter (max of all intensities within patch)
→ Min filter
→ Median filter

* Bilateral Filtering

→ Location specific filter
→ Retain edges and complex structure, but smoothen other areas.

Lecture-4 Human Eye and Color Perception

* Mechanism of human eye

* Human eye capability

* The retina
  → There are many rodes than cones.

* Blind spot
  → Try ~~yourse~~ yourself.

* LMS cones and rodes
→ For low light - active
→ cones are active if enough light.
→ LMS - wave length : Large, Medium, High

→ LMS donot percieve color. Brain does

* scotopic vision : when observer sees stimulus of illumination between 10^{-3} cd/m^2 and 3×10^{-10} cd/m^2

* Weber's law

→ cones are more sensitive towards illumination
→ rods are less sensitive.

* Mach band

* Luminous efficiency
  → Purkinje shift : red seems darker in less light.

* Trichromatic theory

Lecture-5 Geometric Operations

* Image transformation

* Application

* Geometric operations are about moving pixels from one position to other. $I(x,y) \rightarrow I'(x', y')$

* Translation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} dx \\ dy \end{pmatrix}$$

* Scaling

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} Sx & 0 \\ 0 & Sy \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

* Shearing

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & b_x \\ b_y & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

shearing along x and y axis by b_x and b_y

* Rotation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

* Flipping

vertical flip $\begin{pmatrix} r \\ c \end{pmatrix} \Rightarrow \begin{pmatrix} m-r \\ c \end{pmatrix}$

horizontal flip $\begin{pmatrix} r \\ c \end{pmatrix} \Rightarrow \begin{pmatrix} r \\ n-c \end{pmatrix}$

* Image warping

$r \longrightarrow s \qquad T(r) = s$

$s \longrightarrow r \qquad T^{-1}(s) = r$

→ From output pixel, get the related input pixels and interpolate them.

→ Gaussian interpolation

→ Bilinear interpolation

Scanned by CamScanner

* Rotation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$\boxed{\text{or}}$

$$x' = \cos\alpha \, x - \sin\alpha \, y$$
$$y' = \sin\alpha \, x + \cos\alpha \, y$$

$\rightarrow \cos\alpha \, x' + \sin\alpha \, y' = x(\cos^2\alpha + \sin^2\alpha) = x$

$\rightarrow (\cos\alpha) y = y' - \sin\alpha \cos\alpha \, x' + \sin^2\alpha \, y'$

$(\cos\alpha) y = \cos^2\alpha \, y - \sin\alpha \cos\alpha \, x'$

$y = \cos\alpha \, y - \sin\alpha \, x'$

* Homogeneous Co-ordinates and affine transformation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

* Affine transformations

→ Preserves collinearity
(all points in one line remains in one line)
(parallel lines remains parallel)

→ 6 degrees of freedom

* 2D projective transformation (homography)

→ Preserves collinearity but parallel lines don't remain parallel.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

→ 8 degrees of freedom
→ Homography estimation for stereo matching

* Transformation categories

* Non-planar
→ Lines don't remain lines.

* Ripple
* Twirl
* Spherical

Lecture-6 Image Resampling

* Image down sampling
* Gaussian Pyramids
* Image up sampling

* **Aliasing**

→ Sampling rate is not enough
→ Sampling rate ≥ 2 * max frequency in image
→ Minimum sampling rate is called Nyquist rate (on the basis of sampling theorem)

⇒ Down sampling with Gaussian pre-filtering

⇒ Gaussian pyramid

* Laplacian from Difference of Gaussian (DoG)

* Laplacian of Gaussian (LoG)

* Blending

$$I_c = \sum_{i=0}^{l} C_i \qquad C_i = L_i^F M_i + L_i^B (1 - M_i)$$

$$L_i = G_i - G_{i+1} \qquad L_0 = I$$

where I = original image
G_i = gausian at different levels
L_i = laplacian at different levels
C_i = composite of image, mask, etc.

* Up sampling the image with interpolation

→ NN interpolation
→ linear interpolation
→ bilinear interpolation
→ cubic → Quadratic → Sinc → gaussian
→ tent (triangle in 2D - 2D linear)
→ Bicubic

Lecture-7 Color Image Processing

* Attributes of color
* Primary colors
* Lab color space
* Other color space
* Pseudo color image processing
* Applications

* Related and Unrelated color

→ Hue :- Which color from spectrum is this.
Purity of color.

→ Achromatic and Chromatic color

→ Colorfulness → Illumination affects colorfulness.

→ Brightness

→ Saturation : colorfulness of an area judged ~~is~~ to its brightness.

→ Lightness :

→ Chroma :

* Primary colors

* rg color space.

* Color matching functions

* XYZ color space

* (x,y) chromaticity diagram (Yxy)

* Other color spaces

* LMS color spaces

* White balancing
  - Von Kries Method

Lecture-8

In other notebook

Lecture-9

* Correlation
* Convolution

* Convolution operator

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

$$\omega(x,y) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s,t)\, f(x-s, y-t)$$

* Dirac's delta function

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{else} \end{cases}$$

$$\int_{-\infty}^{\infty} f(x)\,\delta(x)\,dx = \int_{-\infty}^{\infty} f(x)\,\delta(x)\,dx = f(0)$$

* $f(t) * h(t) = \int_{-\infty}^{\infty} f(\tau)\, h(t-\tau)\, d\tau$

$h(t-\tau) = h(t) * \delta(t-\tau)$

$= \int_{-\infty}^{\infty} h(T)\, \delta(T-(t-\tau))\, dT$

$F(f(t) * h(t)) = \int_{-\infty}^{\infty} (f * h)(t)\, e^{-2\pi j u t}\, dt$

$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau)\, h(t-\tau)\, d\tau\, e^{-2\pi j u t}\, dt$

$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau) \int_{-\infty}^{\infty} h(T)\, \delta(T-(t-\tau))\, dT\, d\tau\, e^{-2\pi j u t}\, dt$

$t - \tau = K \quad \therefore \quad t = K + \tau$

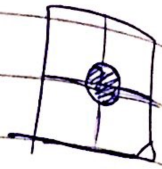
$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau) \int_{-\infty}^{\infty} h(T)\, \delta(T-K)\, dT\, d\tau\, e^{-2\pi j u (K+\tau)}\, dK$

$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau) \underline{\int_{-\infty}^{\infty} h(T)\, \delta(T-K)}\, e^{-2\pi j u K}\, e^{-2\pi j u \tau}\, \underline{dT}\, d\tau\, dK$

$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau)\, h(K)\, e^{-2\pi j u K}\, e^{-2\pi j u \tau}\, d\tau\, dK$

$= F(u) \cdot H(u) = \left(\int_{-\infty}^{\infty} f(\tau)\, e^{-2\pi j u \tau}\, d\tau\right)\left(\int_{-\infty}^{\infty} h(K)\, e^{-2\pi j u K}\, dK\right)$

* Low pass filter (Blurring)



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

→ Due to sharp low pass filter, '**ringing effect**' appears.

* $$x(t) = \begin{cases} A & -w/2 \le t \le w/2 \\ 0 & \text{else} \end{cases}$$

$$X(u) = \int_{-\infty}^{\infty} x(t)\, e^{-2\pi j u t}\, dt$$

$$= A \int_{-w/2}^{w/2} e^{-2\pi j u t}\, dt$$

$$= \frac{A}{(-2\pi j u)} \left[e^{-2\pi j u t} \right]_{-w/2}^{w/2}$$

$$= \frac{A}{2\pi u}(j) \left(e^{-\pi j u w} - e^{\pi j u w} \right)$$

$$= \frac{j A}{\cancel{2}\pi u} \; -\cancel{2} j \sin(\pi u w)$$

$$= \frac{A}{\pi u} \sin(\pi u w)$$

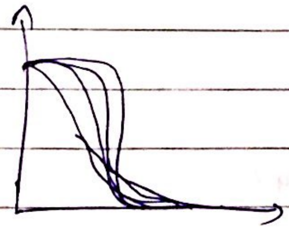
$$= wA \, \frac{\sin(\pi u w)}{\pi u w}$$

$$\boxed{X(u) = wA \, \mathrm{sinc}(\pi u w)}$$

* Butterworth low pass filter

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

where $D(u,v)$ is the distance of u,v from center.



increasing n will give more like previous ideal low pass filter.

* Gaussian low pass filtering

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

* High pass filter (Image sharpening)

$$\underbrace{H'(u,v)}_{\text{(high pass)}} = 1 - \underbrace{H(u,v)}_{\text{(low pass)}}$$

→ $H_{Hp}(u,v)$ for high pass filter.

→ Image to fourier image $F(u,v)$ ...

$$F^{-1}\left(\left(1 + H_{Hp}(u,v)\right) F(u,v) \right)$$

$$= \text{original image} + \text{edges}$$

* Laplacian in fourier domain

$$F\left(\frac{d^n f(x)}{dx^n}\right) = (ju)^n \; F(u)$$

$$f(x) \longrightarrow F_1(u)$$

$$\therefore \; F_1(u) = \int_{-\infty}^{\infty} f(x) \, e^{-2\pi j u x} \, dx$$

$$\rightarrow \quad f'(x) \longrightarrow F_2(u)$$

$$\cancel{F_2(u) = \int_{-\infty}^{\infty} f'(x) \, e^{-2\pi j u x} \, dx}$$

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(u) \, e^{+2\pi j u x} \, du$$

$$f'(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(u) \, e^{+2\pi j u x} \, (2\pi j u) \, du$$

$$f'(x) = \int_{-\infty}^{\infty} F_1(u) \, e^{+2\pi j u x} \, (+ju) \, du$$

$$f'(x) = \int_{-\infty}^{\infty} \left(+ju \, F_1(u)\right) e^{+2\pi j u x} \, du$$

$$\therefore \; f'(x) \longrightarrow ju \, F_1(u)$$

$$\therefore \; F\left(f^n(x)\right) = (ju)^n \, F(u)$$

* Notch filter

→ It is used to remove repeatitive spectral noise.