2 Image

Monday, 19 January 2026 16:12

image: continuous func. over space

→ func gives brightness level or color info

→ most images are rectangle

→ img's are continuous in amplitude

→ " " " in space = no pixels / grid

Sampling:

quantization:

pixel: pic. element represents brightness @ one point

$f(x, y) = f_{yx}$

x: col

y: row (index)

$$f = \begin{bmatrix} f(0,0) & f(1,0) & \dots & f(N-1,0) \\ f(0,L-1) & & & \end{bmatrix}$$

(→ x, ↓ y)

Color imgs: $R(x,y)$, $G(x,y)$, $B(x,y)$

monochrome → above equals

Storage

- $L \times N$ pixels
- 2^B gray levels
- c color components

} Size $= L \times N \times B \times c$

Ex $L = N = 512$

$B = 8$

$c = 1$ → monochrome

Size $= 512 \times 512 \times 8 \times 1$

$= 2097152$ bits

$\frac{2097.152 \text{ bits}}{8} = 262.144$ bytes

$\frac{262.144 \text{ bytes}}{1024} = 256$ kBytes

Ex $L \times N = 1024 \times 1280$

$B = 8$

$c = 3$

$S = 1024 \times 1280 \times 8 \times 3$

$= 31.457.280$ bits

$= \frac{31.457.280}{8}$ bytes

$= \frac{31.457.280}{8} \cdot \frac{1}{1048.576}$ MB

$= 3.75$ mb

→ for video

4K: $2160 \times 3840 \times 8 \times 3 \times 50$

- 2160×3840: pixels / frame
- $\times 8 \times 3$: bits / frame
- $\times 50$: bits @ 50 fps

$= 9.953.280.000 \approx 10^{10}$ bits ≈ 10 Gb/s

$= 9.953.280.000 / 8 \frac{\text{bytes}}{s} = 1.2$ Gb/s

I: luminance $\frac{cd}{m^2}$

: physical measure related to perceived brightness

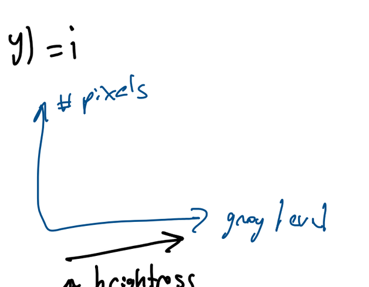
visibility threshold $= \frac{\Delta I}{I} \approx 1 \dots 2\ \%$

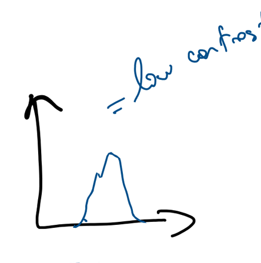
↳ weber's Law

Histogram: how many pixels fall into each intensity value

1) For B bits image, init 2^B counters w/ 0
2) Loop for all pixels → x, y
3) increment #i, $f(x, y) = i$

histogram = pdf





! histogram ignores where these pixels are

Histogram Equalization: if img uses small part of [0,255], it has low contrast. So, equalization is required. (non-linear func.)

$g = T(f)$ $\quad 0 < f < 1$, $0 < g < 1$

goal: $p_g(g) =$ const.

↳ pdf

in continuous:

PDF: $p_g(g) = \left[p_f(f) \frac{df}{dg} \right]_{f = T^{-1}(g)}$

CDF: $g = T(f) = \int_0^f p_f(\alpha)\, d\alpha$ (CDF) $\quad 0 \le f \le 1$

↳ so $0 \le g \le 1$

Diff g:

$\frac{dg}{df} = p_f(f)$

$p_g(g) = \left[p_f(f) \frac{df}{dg} \right]_{f = T^{-1}(g)} = \left[p_f(f) \frac{1}{p_f(f)} \right]$

$= 1$

↳ output dist. = uniform / constant over [0,1]

discrete case:

8 bit, $L = 256$

↳ $f: f_0, f_1, \dots, f_{L-1}$

$p_0 = \frac{n_0}{n}$, $p_1 = \frac{n_1}{n}$, …, $p_{L-1} = \frac{n_{L-1}}{n}$

- n_1 → # pixels having value 1
- n → # pixels
- p_1 → Pmf

CDF: $g_k = T(f_k) = \int_0^f p_f(\alpha)\, d\alpha$

$= \sum_{i=0}^{k} p_i$

After Equalization:

- Dark → Darker; Bright → Brighter
- histogram values are spreaded out across range

Adaptive HE

- improves contrast locally
- handles where contrast varies across the scene

1) Sliding Window Approach
- local histogram, local mapping
- expensive

2) Tiling Approach
- split to tiles
- one mapping / tiles
- blend mappings

- Clipping is needed → Local HE amplifies noises in flat areas.

- Small tiles → more local contrast (+ noise risk)
- larger tiles → " smooth
- global histogram → over / under enhance

Point Operations Btw. Images

1) Noise Cancellation

- Average multiple captures → ⇓ noise
- preserve content
- get clearer pic
- ↑ input captures → better

N images: $f_1(x,y)$, $f_2(x,y)$, …, $f_N(x,y)$

Avr img's: $\overline{f(x,y)} = \frac{1}{N} \sum_{i=1}^{N} f_i(x,y)$

MSE: $E\{(\bar{f} - g)^2\}$

$= E\left\{\left(\left(\frac{1}{N} \sum f_i\right) - g\right)^2\right\}$

$= E\left\{\left(\left(\frac{1}{N} \sum (g + n_i)\right) - g\right)^2\right\}$ (n_i → noise in image i)

$= E\left\{\left(\frac{1}{N} \sum_i n_i\right)^2\right\}$

$= \frac{1}{N^2} \sum E\{n_i^2\} = \frac{1}{N} E\{n^2\}$

Averaging reduces noise by $\frac{1}{N}$

2) High Dynamic Range Imaging

16 exposures → large brightness range

- some img's capture highlights
- " " " shadows

One f-stop → 2 in exposure (double / half light capturing)

HDR combination = select pixels from each exp. ⊕ merge in HDR ⊕ tone mapping

3) Image Subtraction

→ $d(x,y) = |f_1(x,y) - f_2(x,y)|$

→ change = light values