Histogram Equalization 1/21/2020

load image

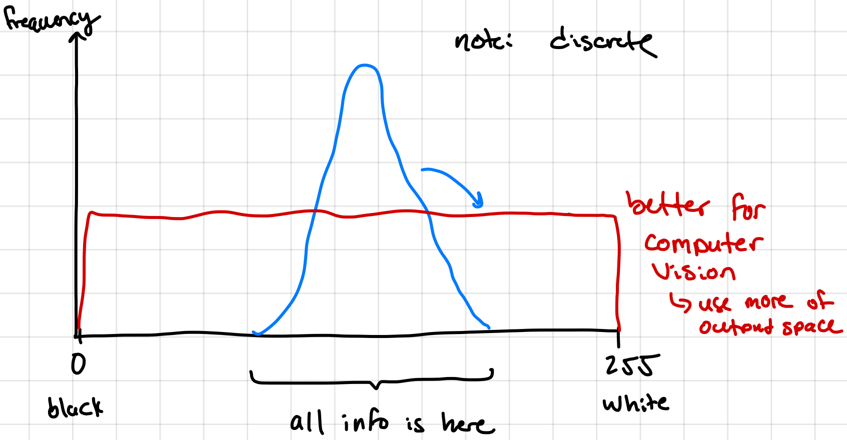
```
plt.hist(
        img[:,:,1].ravel()
            R  C  color
                  = G
        bins = list(range(256))
    )
```

Histogram equalization → uniform dist. target

… matching → arbitrary dist. target (Gaussian, etc.)

print(gray.shape) → (X, y)
(256, 256)

color → grayscale is a LC of R, G, B channels



note: discrete

- note lose spatial info in histogram
- poor contrast here

- point process to change histogram distribution

Pseudocode

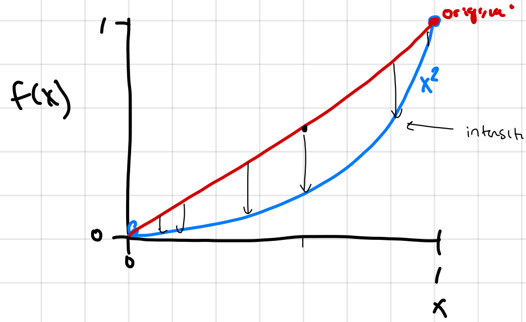
```
gray = Imread(image, cv2.Imread_grayscale)

def point_process(x):
    """ apply pt. process to numpy array """
    return (x/255)**2 * 255
           int → float → int        .astype('uint8')

gray = point_process(gray)
```

- imshow scales things so it fills 255
- like imsc or soundsc in MATLAB
- fix: norm = NoNorm()

from matplotlib.colors import NoNorm



intensities get mapped down in CCG pt processes

Histogram equalization pt. process

x	input intensity	btwn $[0,1]$
$y=f(x)$	output intensity	$[0,1]$
$P_x(x)$	distribution on x	
$P_y(y)$	distribution on $y \simeq c$	

assume continuous for now

const.

$$p(y) = \frac{p(x)}{\left|\frac{df}{dx}\right|}$$

input dist

output dist

(dy/dx)

change of variable

Example:

① If $\therefore$ → output is cdf · constant

up to x

cdf over x

$$y = f(x) = C \cdot \underbrace{P(x \leq X)}$$

$$= C \int_0^x P_x(v)\,dv$$

$$\frac{df}{dx} = \frac{d}{dx}\left[C \int_0^x P_x(v)\,dv\right]$$

$$= C \frac{d}{dx} \int_0^x P_x(v)\,dv$$

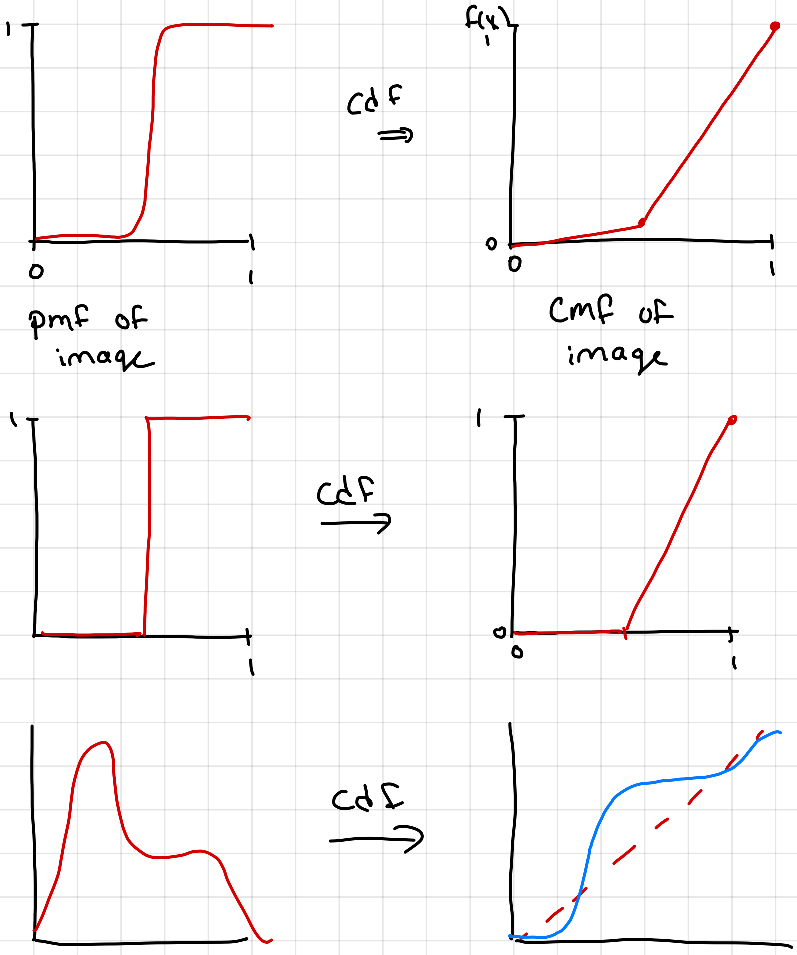
$$\frac{df}{dx} = C\,P_x(x)$$

therefore

$$P_y(y) = \cancel{P_x(x)} \cdot \frac{1}{C\cancel{P_x(x)}}$$

$$P_y(y) = \frac{1}{C}$$

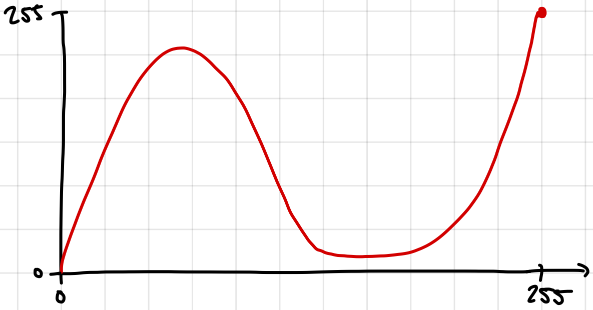
② then ↳ output dist is a constant



output is cdf
↳ equalizes distribution

image → pdf → $f(x) = c \cdot P(x \leq X)$ → equalized hist.
cdf

Some problems



pt. process so all pts get processed the same
↳ exposure issues
↳ sunlight gets pushed down, no room for shadows to pull up

soln → adaptive histogram equalization