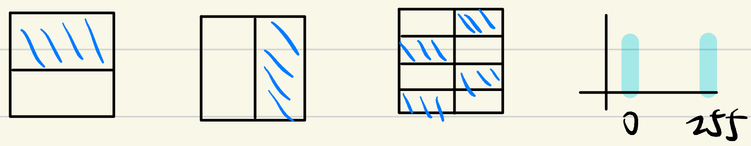
Histogram and Point operation

$I(x,y)$

$h(D)$ = # of pixels in $I(x,y)$ that have intensity D

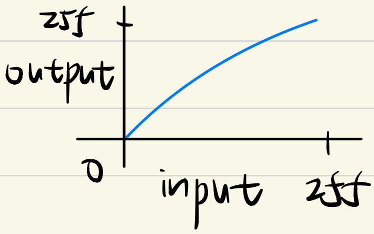
$$= \left|\{I(x,y) \mid I(x,y) = D\}\right|$$

Many Images have the same histogram



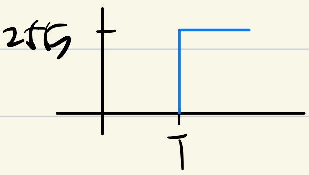
$J(x,y) = F(I(x,y))$

After before



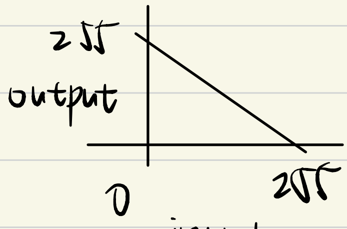
threshold

$F(D) = step(D-T) * 255$

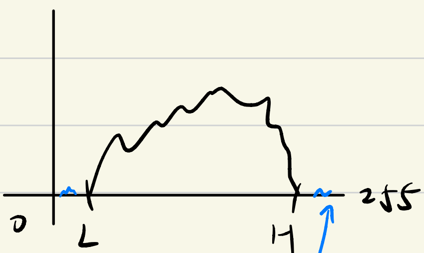


7:46

Digital Negative



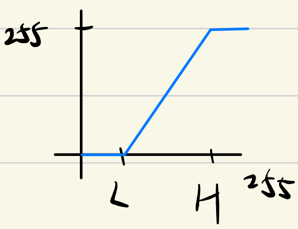
Contrast Stretching



$$F(D) = \frac{255}{H-L}(D-L)$$

realistically, clip to [0, 255]

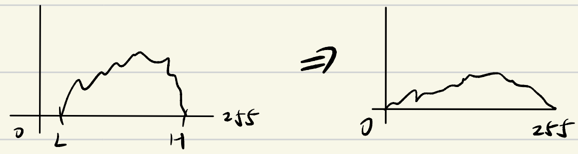
in case of little warps



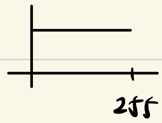
rgb2gray → imtool(im)

non-linear mapping of intensity

histogram equalization

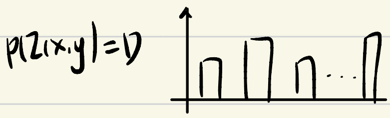


sometimes contrast stretching fails desired histogram:



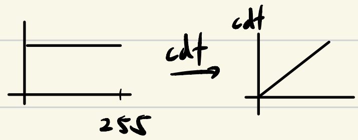
what is $F(D)$?

think of image histogram as a probability mass function



$$\sum_{D=0}^{255} p(I(x,y)=D) = 1$$

we will use the cdf (cumulative df)



what F(D) converts original image histogram into uniform distribution?

turns out that F(D) = C(D)

↑ cdf of original image

$I(x,y) \to J(x,y)$

$$P(J(x,y) \le D) = P(C(I(x,y)) \le D)$$
$$= P(I(x,y) \le C^{-1}(D))$$
$$= C(C^{-1}(D)) = D$$

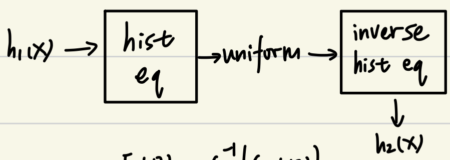
where intensity are scaled in [0,1]

In practice, $F(D) = \dfrac{\sum_{0}^{D} h(D)}{\sum_{x=0}^{255} h(D)}$ → scale to [0,255]

histogram specification

$h_1(D) \to h_2(D)$

old new



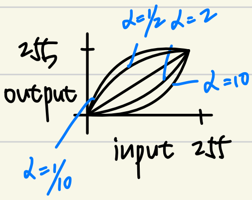
$F(D) = C_2^{-1}(C_1(D))$

gamma correction

every display device has a different non-linear relationship b/t pixel input intensity and display output luminance

The relationship for real device is often modeled as a power funciton

$$F(D) = D^{\alpha}$$



solution: if we know the γ of the display device, pre-compensate intensities:

$$\underbrace{(D^{1/\gamma})^{\gamma}}_{\text{send to display}} = D$$

point operation affect every pixel with the same intensity in the same way

Many image processing operations are more local. eg: spatial filter.
filter an image to replace each pixel by the average of its nieghbours

$$J(x,y) = \frac{1}{9} \sum_{i=-1}^{1} \sum_{j=-1}^{1} I(x-i,\, y-j)$$

1/9	"	"
"	1/9	"
"	"	1/9

$$J[x,y] = \sum_i \sum_j h(x,y)\, I[x-i,\, y-j]$$
$$= h * I \quad \text{convolution of filter and image}$$

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

is an edge detecting spatial filter