Basics of Machine Vision

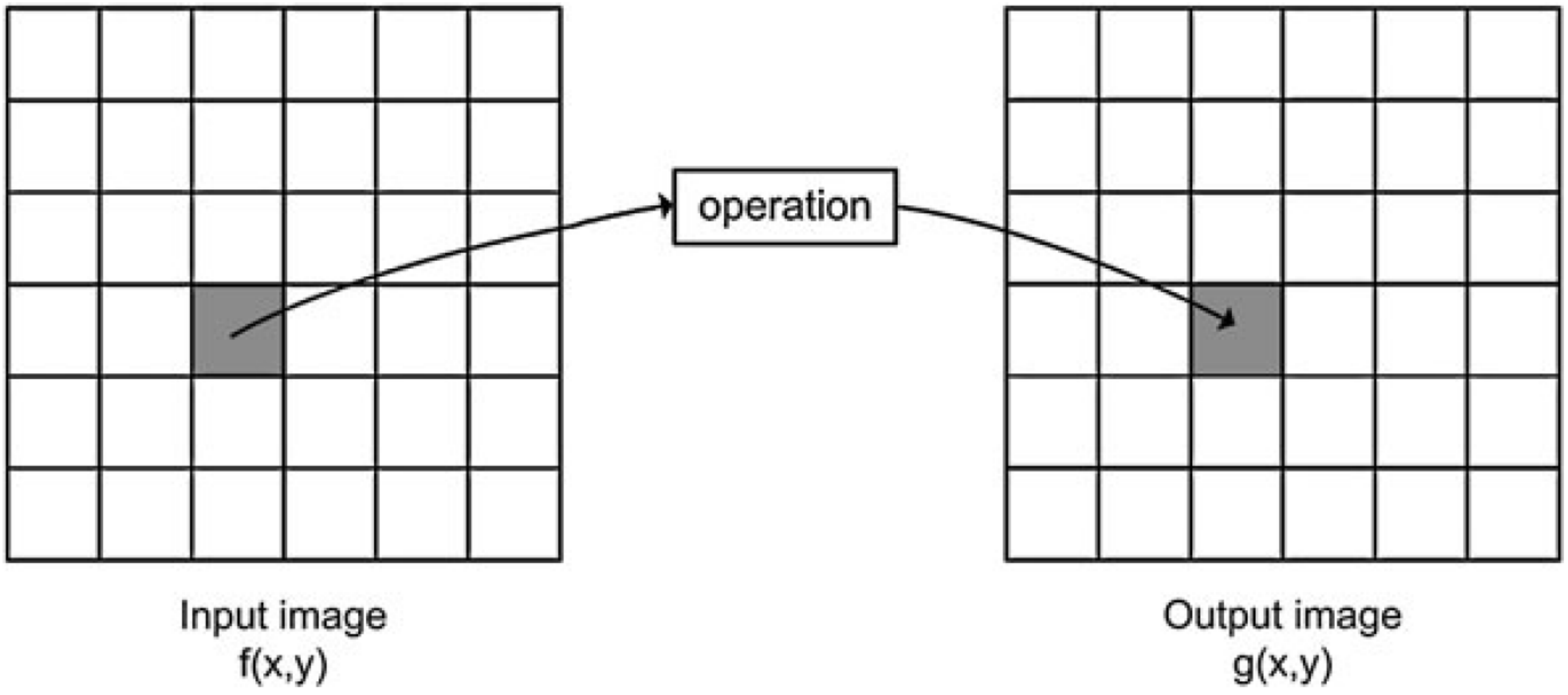
Primož Podržaj

Lecture 02

Two different groups of image processing algorithms

- **Point processing** is defined as an operation which calculates the new value of a pixel based only on the value of the pixel in the same position in the original image.
- **Neighborhood processing** is defined as an operation which calculates the new value of a pixel based on the value of the pixel in the same position in original image as well as on the value of some pixels in the neighborhood.

Point processing in mathematical sense



Example – Gray-Level Mapping

$g(x,y)=f(x,y)+b$ (brightness)

$b < 0$

Decreased brightness

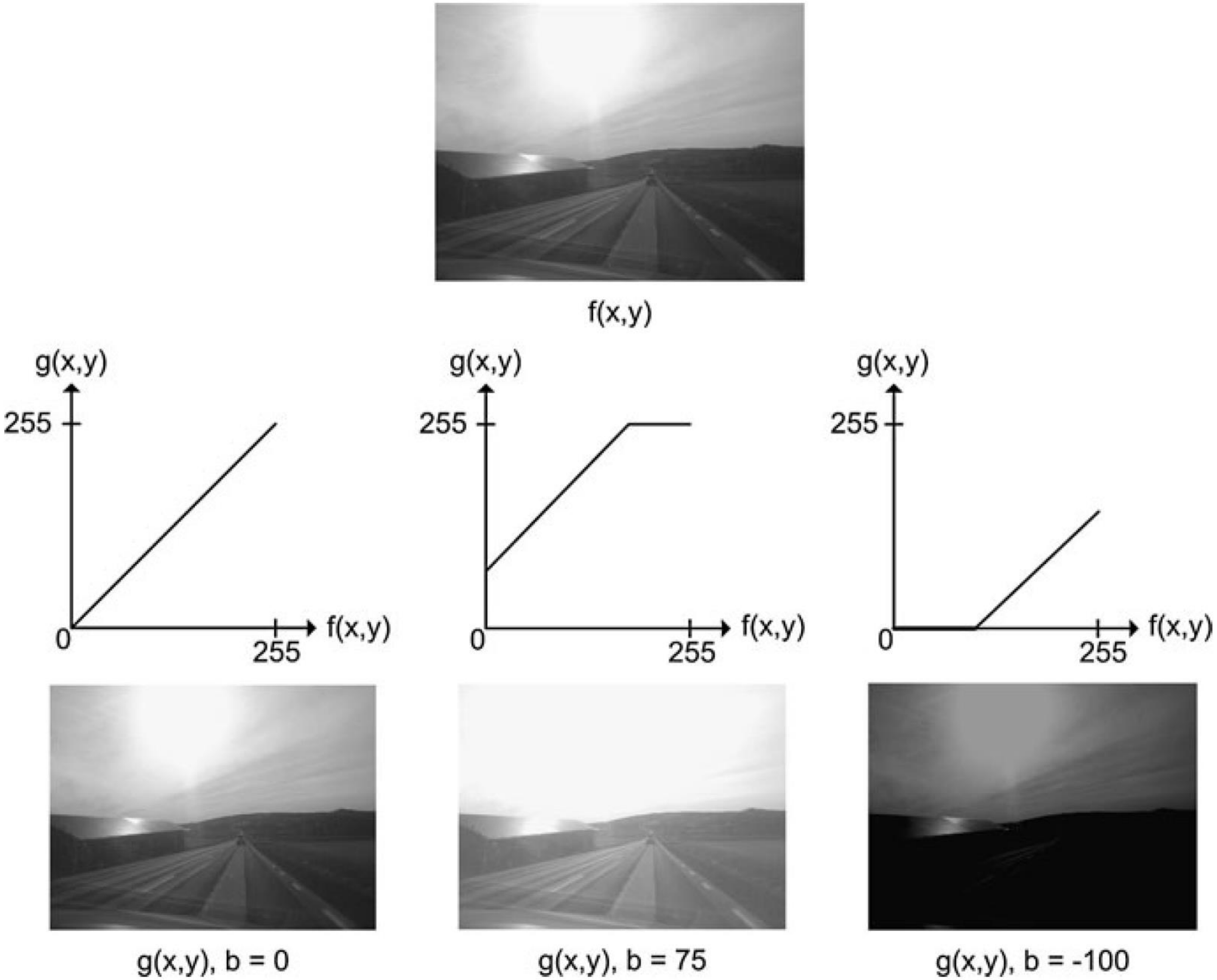
$b = 0$

Input image

$b > 0$

Increased brightness

Limited range



Contrast

g(x,y)=a f(x,y) (contrast)

a < 1 a = 1 a > 1

Decreased contrast Input image Increased contrast

Linear gray-level mapping

$g(x,y)=a\ f(x,y) + b$

In order to get best contrast (values in $[f_1\ f_2]$):

$0=a\ f_1 + b$ and $255=a\ f_2 + b$

we get

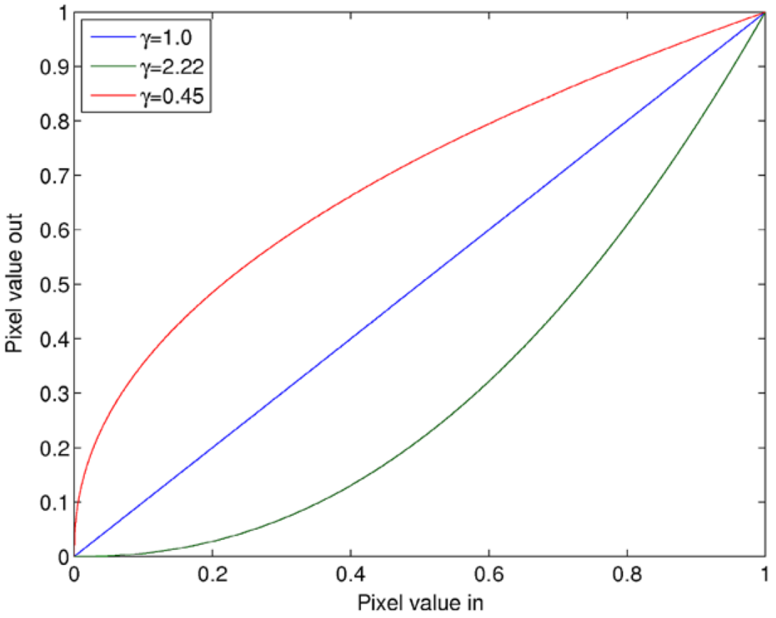
$a=255/(f_2-f_1)$ and $b=-a\ f_1$

Non-linear gray-level mapping

- Gamma mapping
- Logarithmic Mapping
- Exponential Mapping
- Histogram Equalization

Gamma mapping

$g(x,y)=f(x,y)^{\gamma}$ f(x,y) must be mapped to [0,1] first.



Gamma value: 0.45

No gamma correction

Gamma value: 2.22

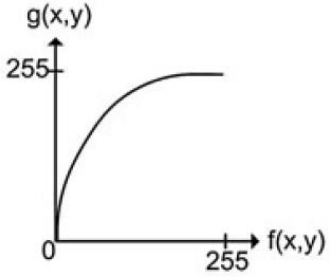
Logarithmic mapping

$g(x,y)=c \log(1+f(x,y))$

$c=255/(\log(1+v_{max}))$

v_{max} is the maximum pixel value in the input image.

f(x,y)



Logarithmic greyscale mapping

g(x,y)

Exponential mapping

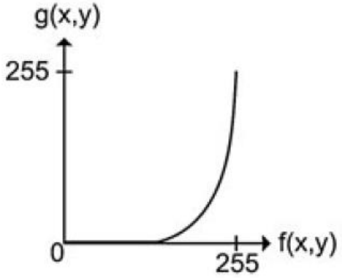
$g(x,y)=c\ (kf(x,y)-1)$

$c=255/(kv_{max}-1)$

v_{max} is the maximum pixel value in the input image.

k is normally chosen as a number just above 1. This will enhance details in the bright areas while decreasing detail in the dark areas

f(x,y)

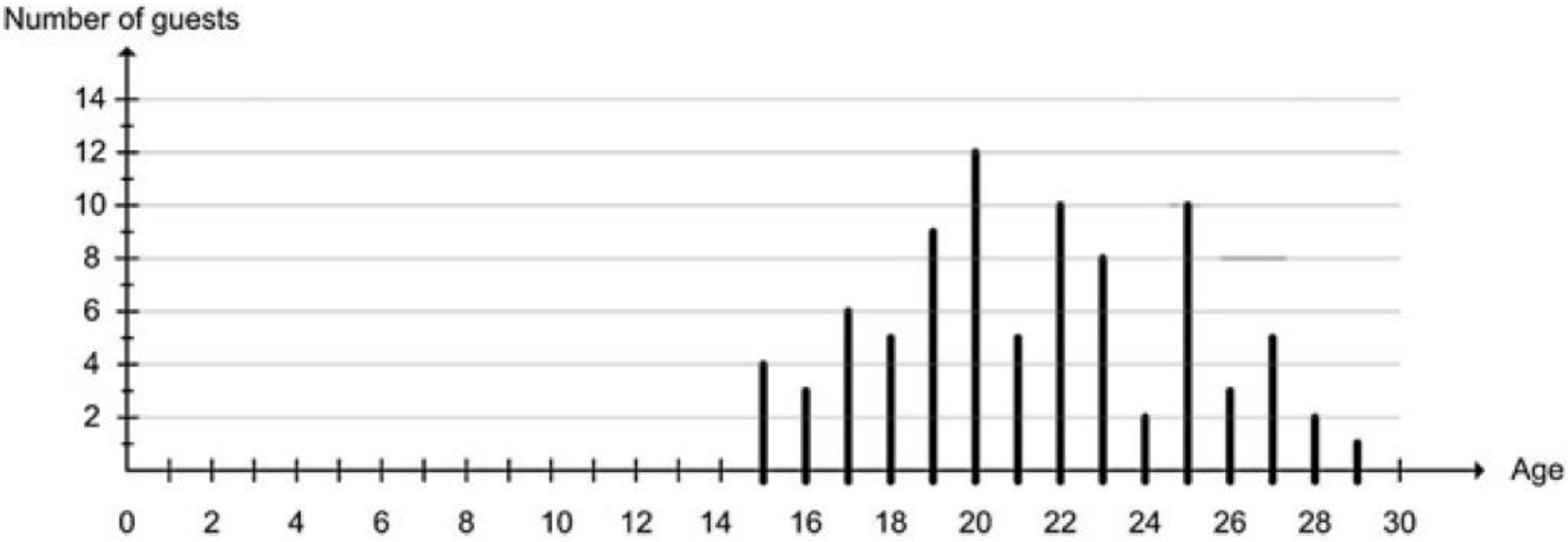


Exponential greyscale mapping

g(x,y)

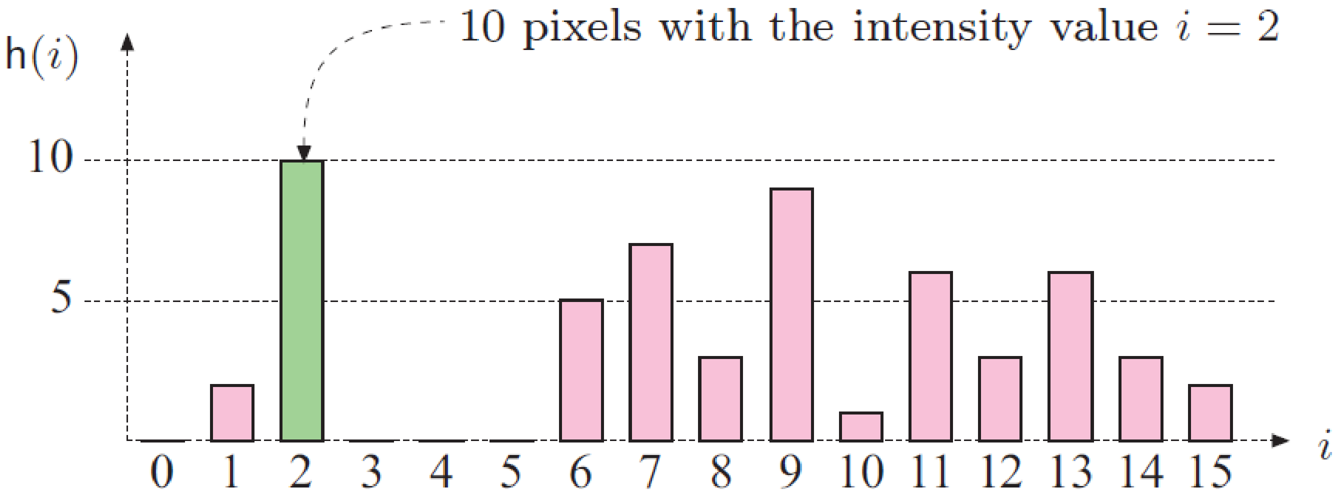
Histogram

- Age of people at a party



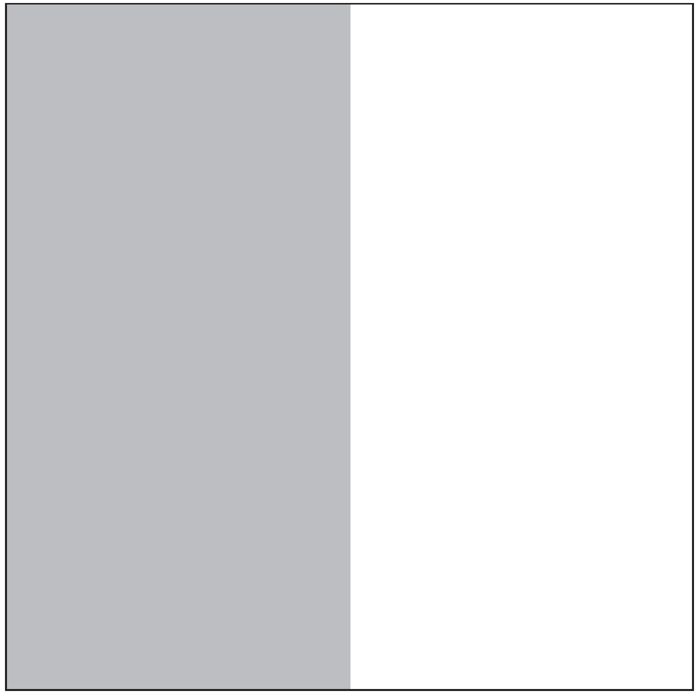
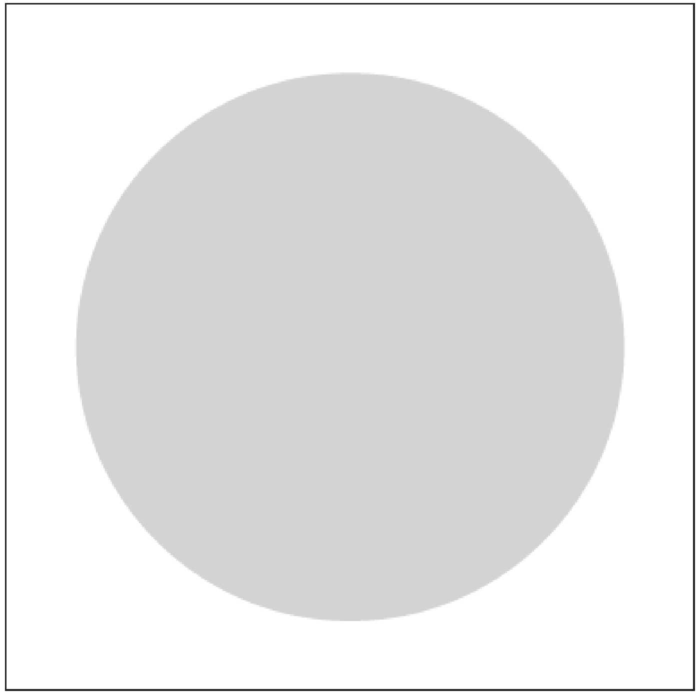
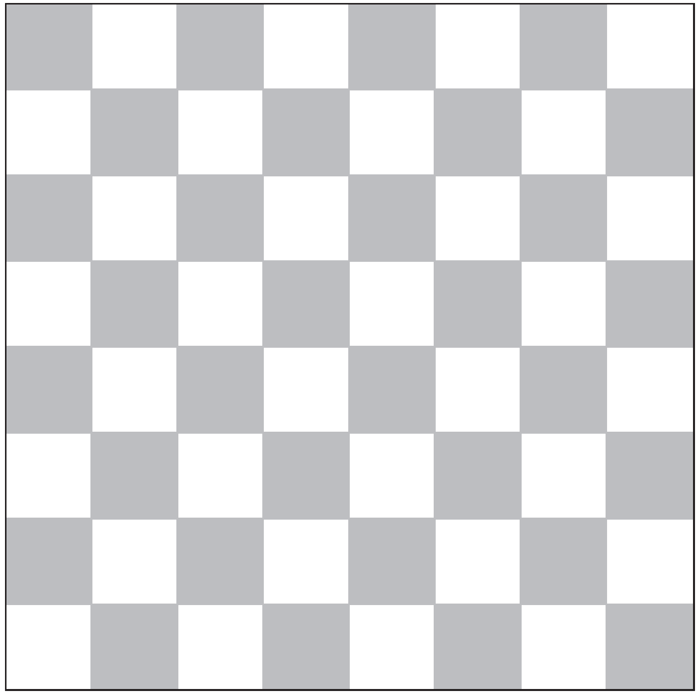
The image histogram

- h(i) = card{ (x,y) | f(x,y)=i }

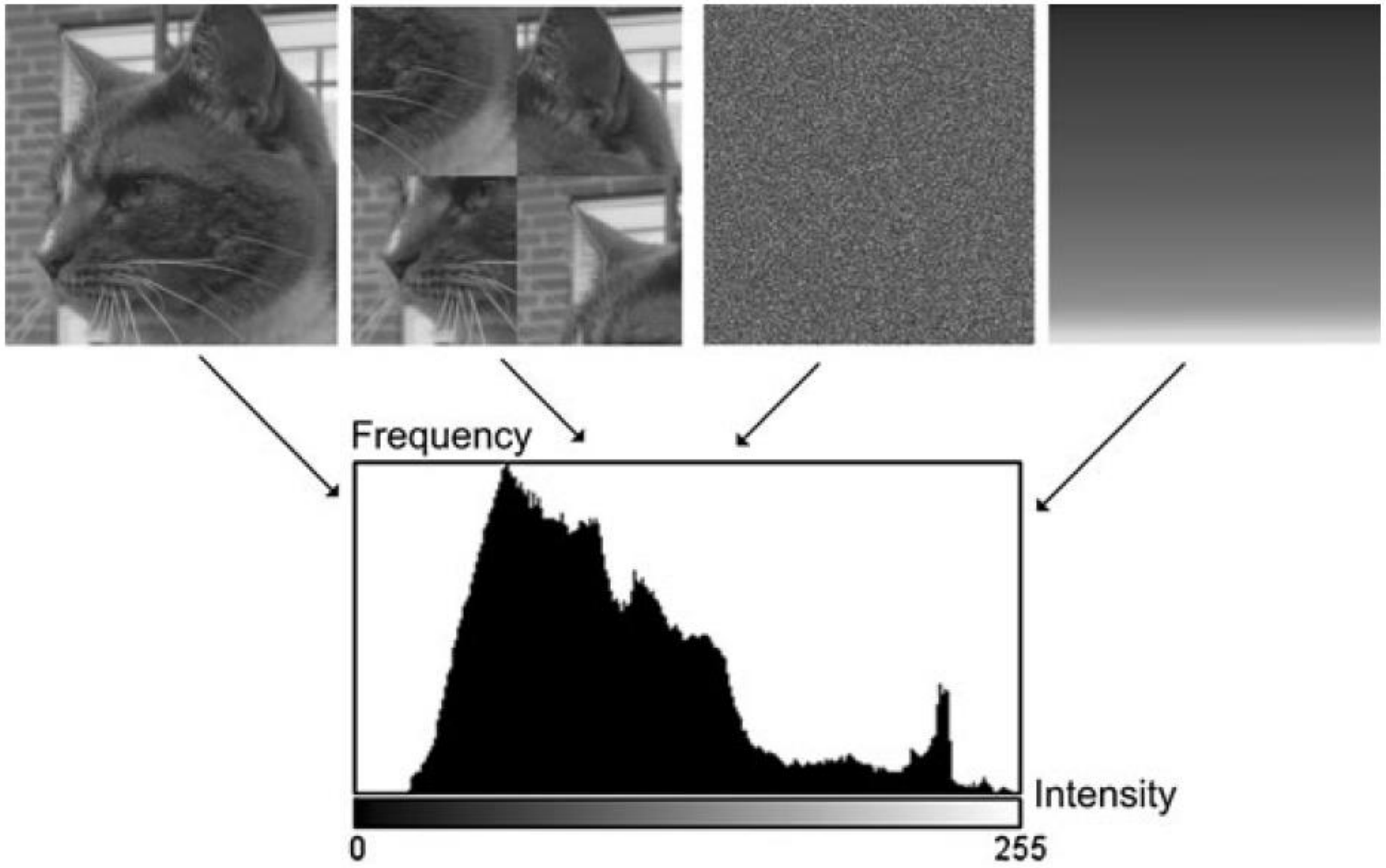


h(i)	0	2	10	0	0	0	5	7	3	9	1	6	3	6	3	2
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

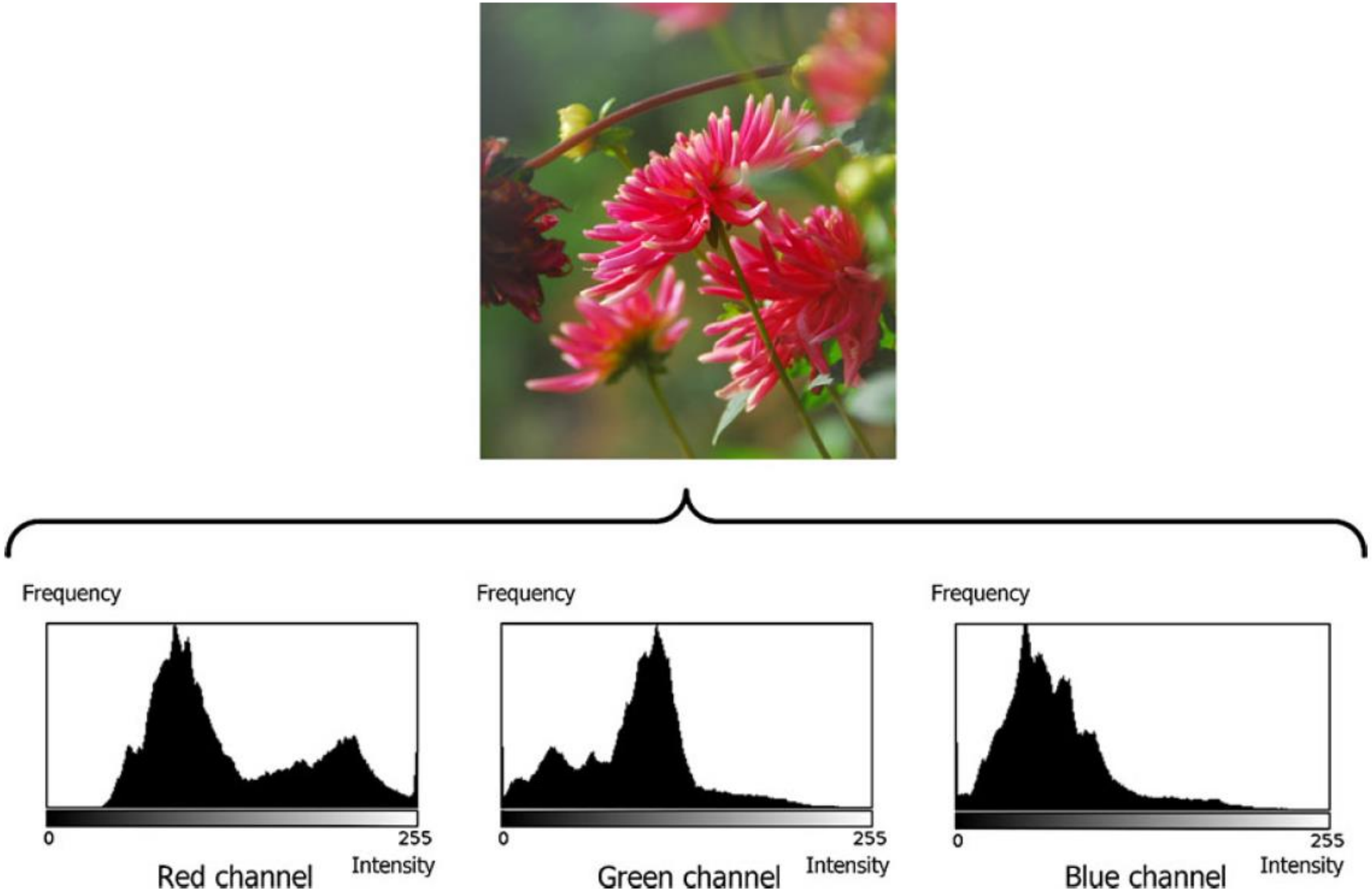
Image ↔ Histogram

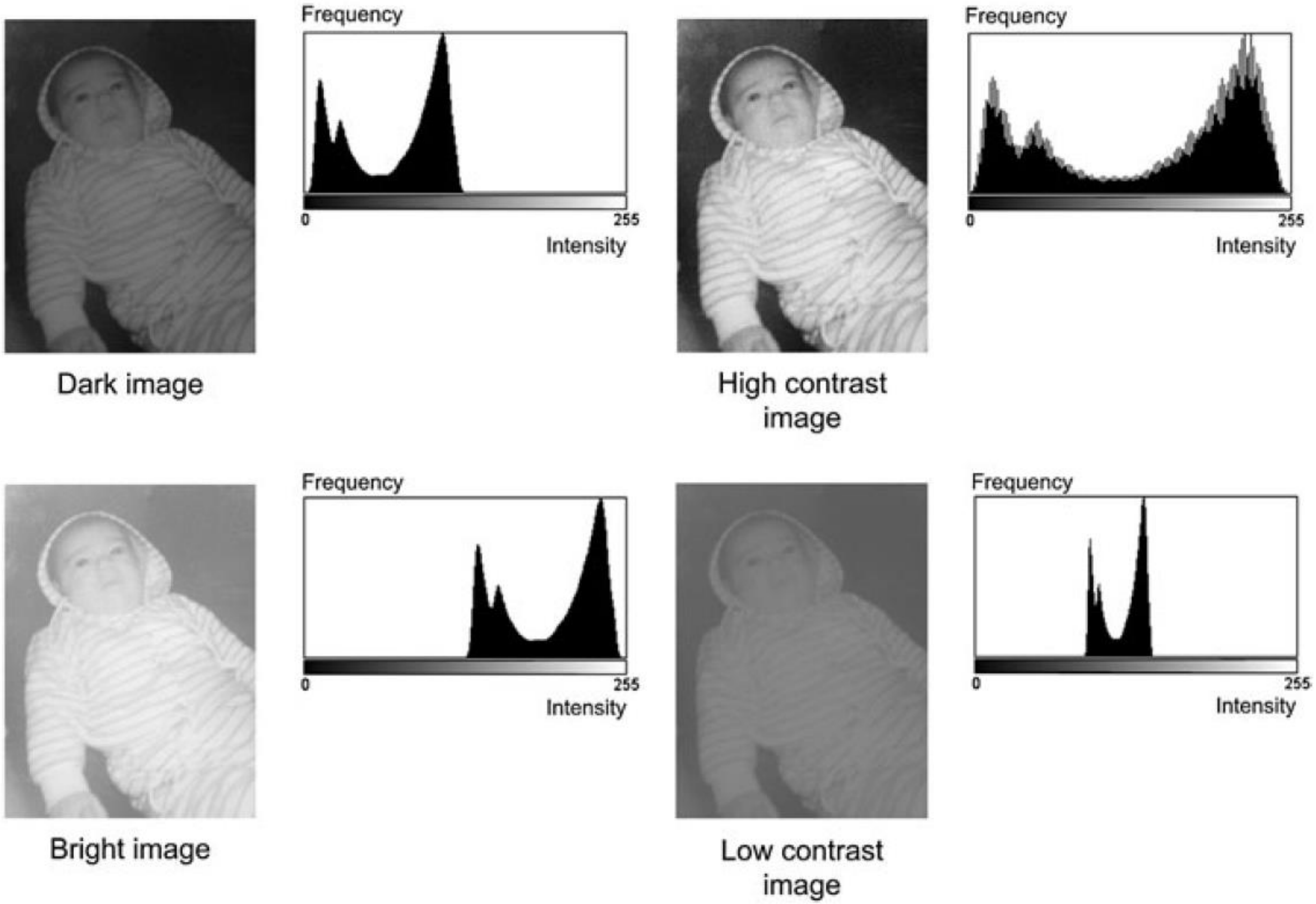
Realation between an image and a histogram



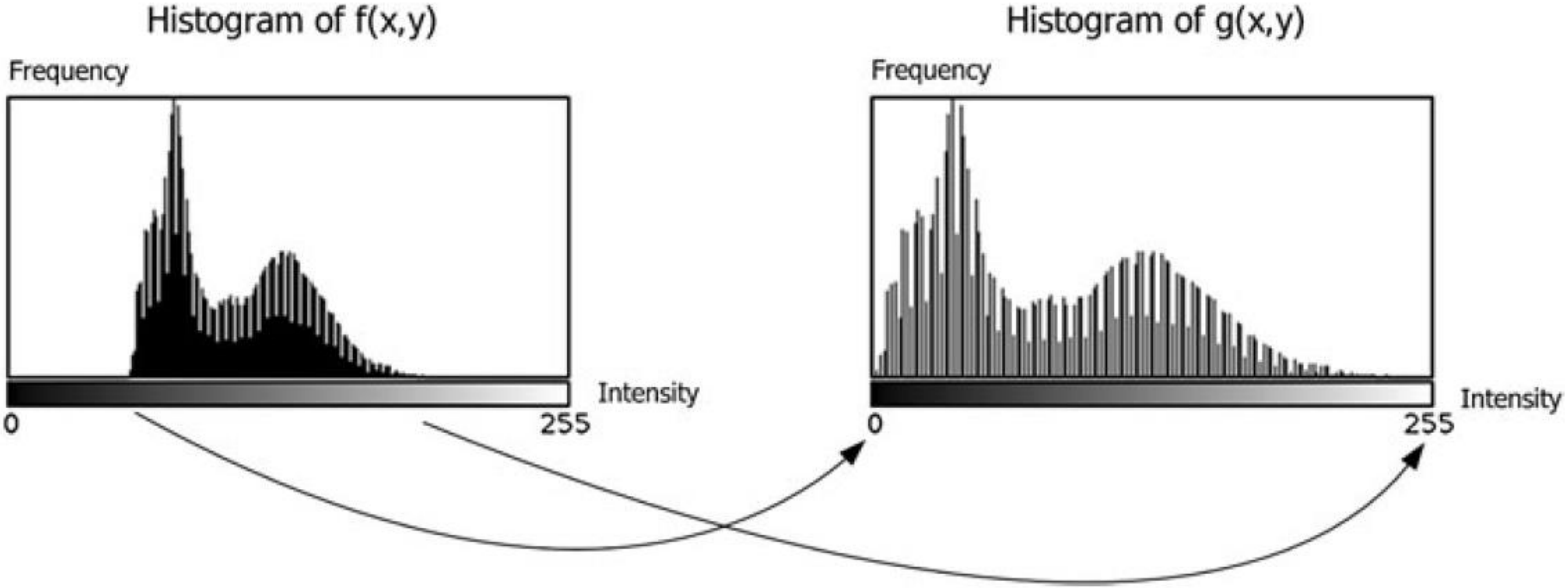
Histogram(s) of a color image



What information can be retrieved from a histogram?

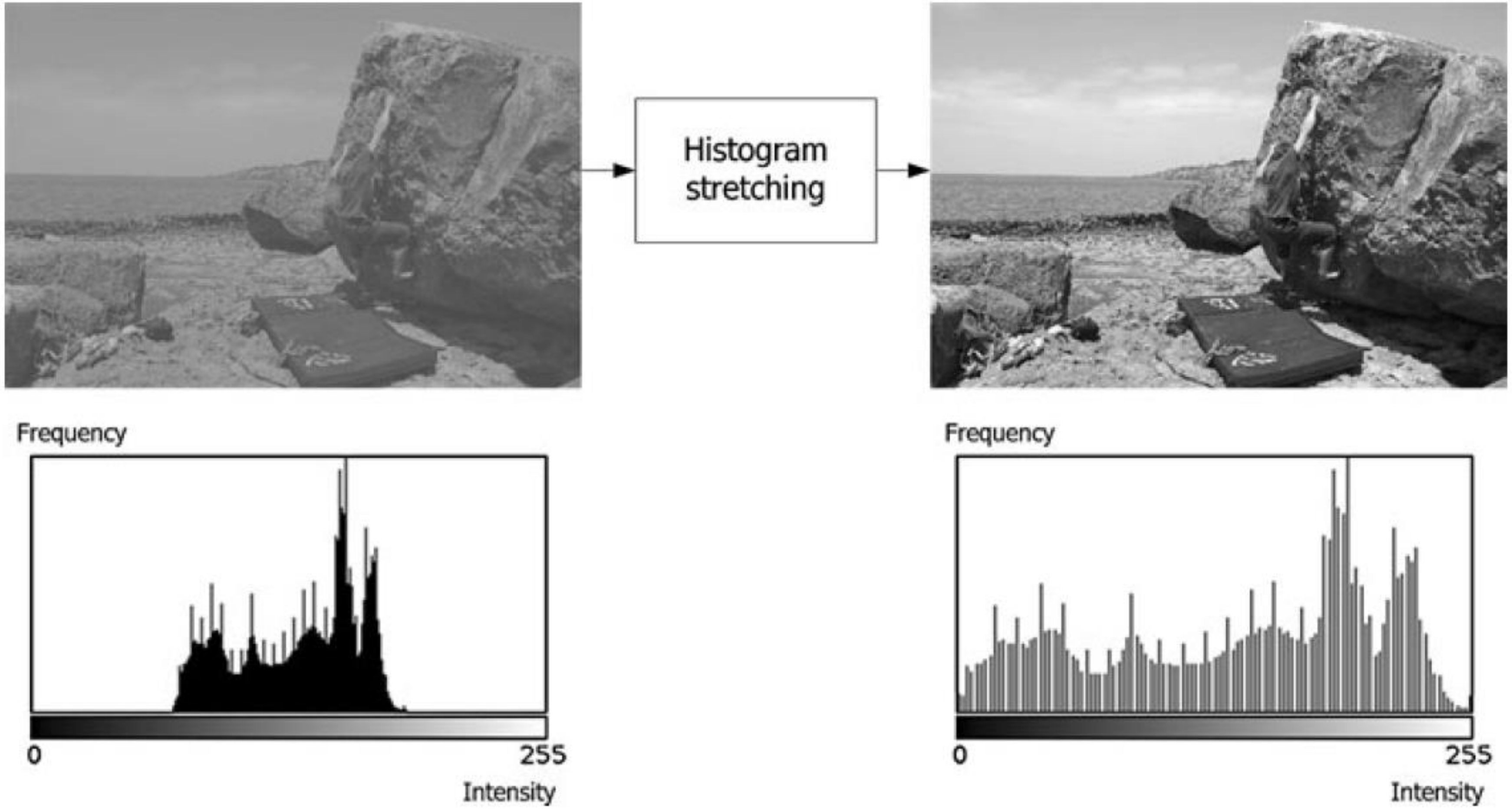


How can an image be enhanced?



Histogram stretching

$g(x,y)=255/(f_2-f_1) * (f(x,y) -f_1)$

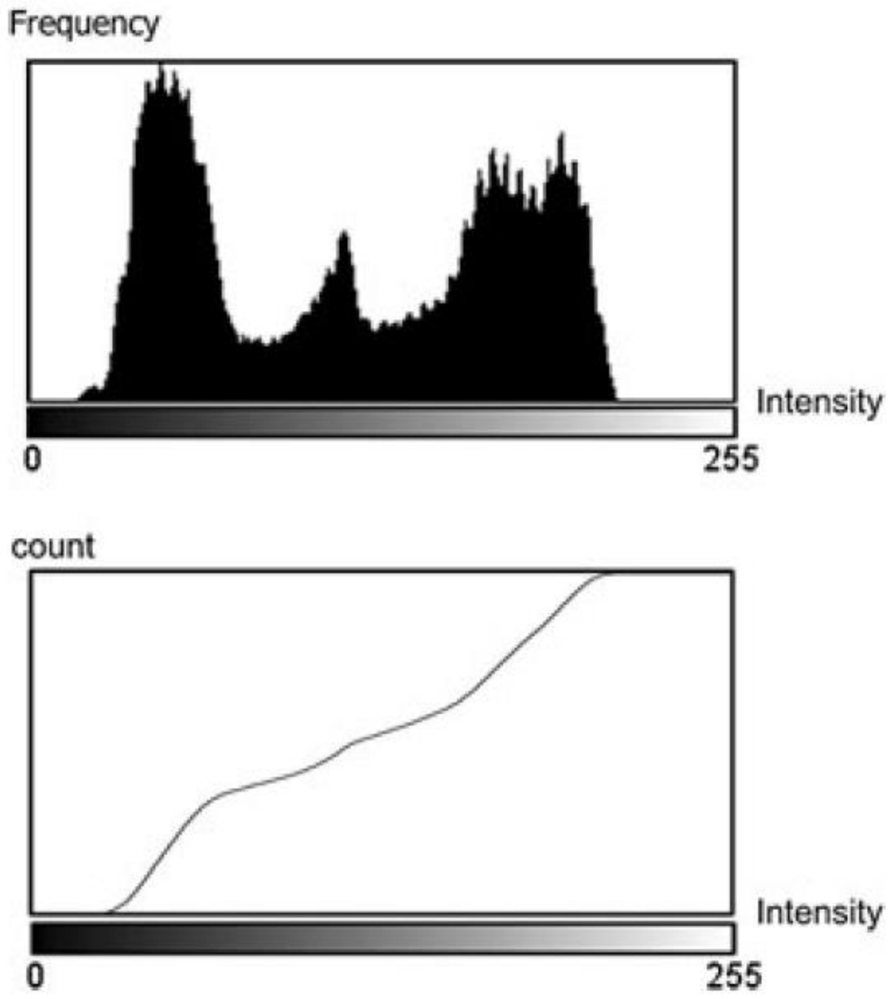


Histogram equalization

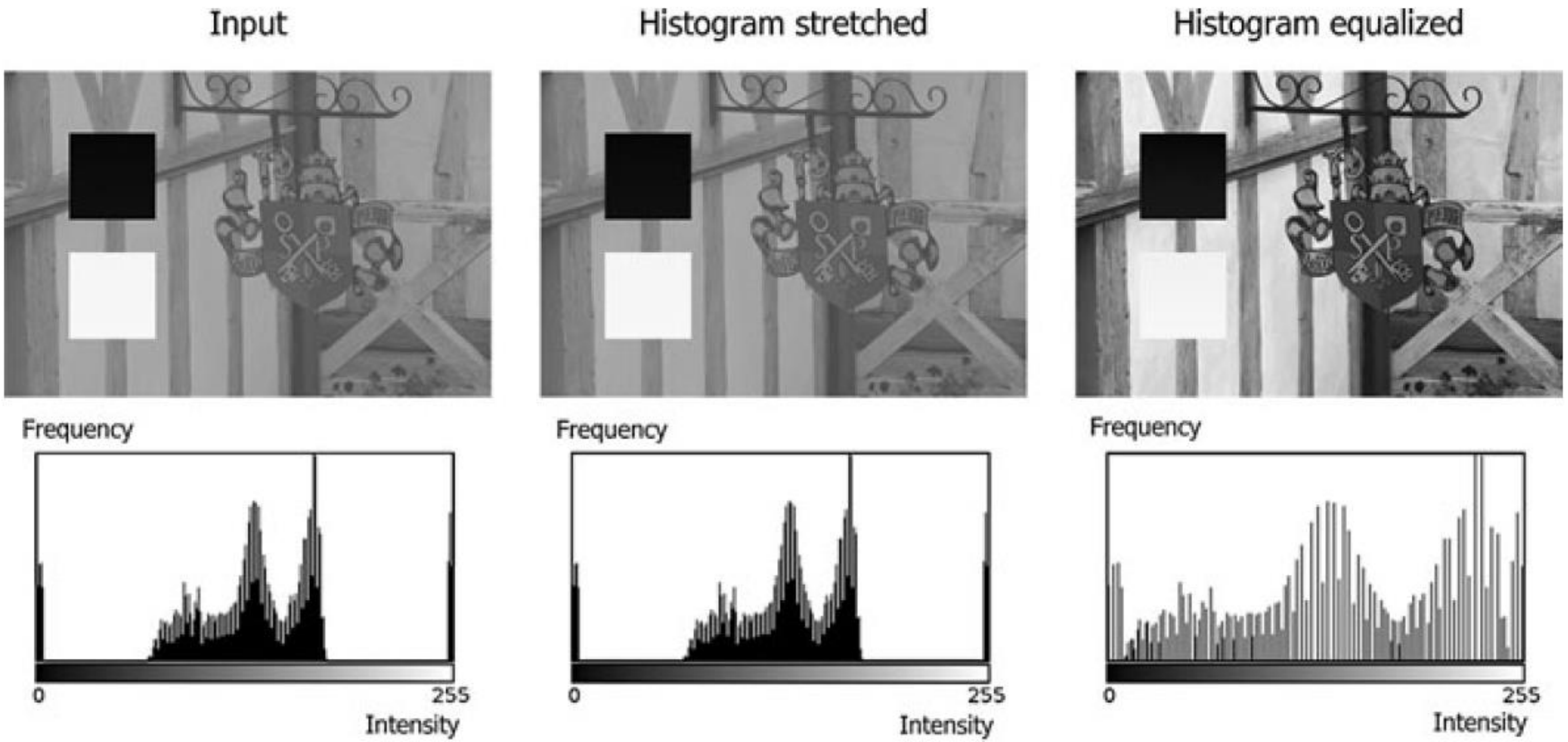
- Cumulative histogram: $C_j = \sum_{i=0}^{j} H[i]$

i	0	1	2	3
$H[i]$	1	5	0	7
$C[i]$	1	6	6	13

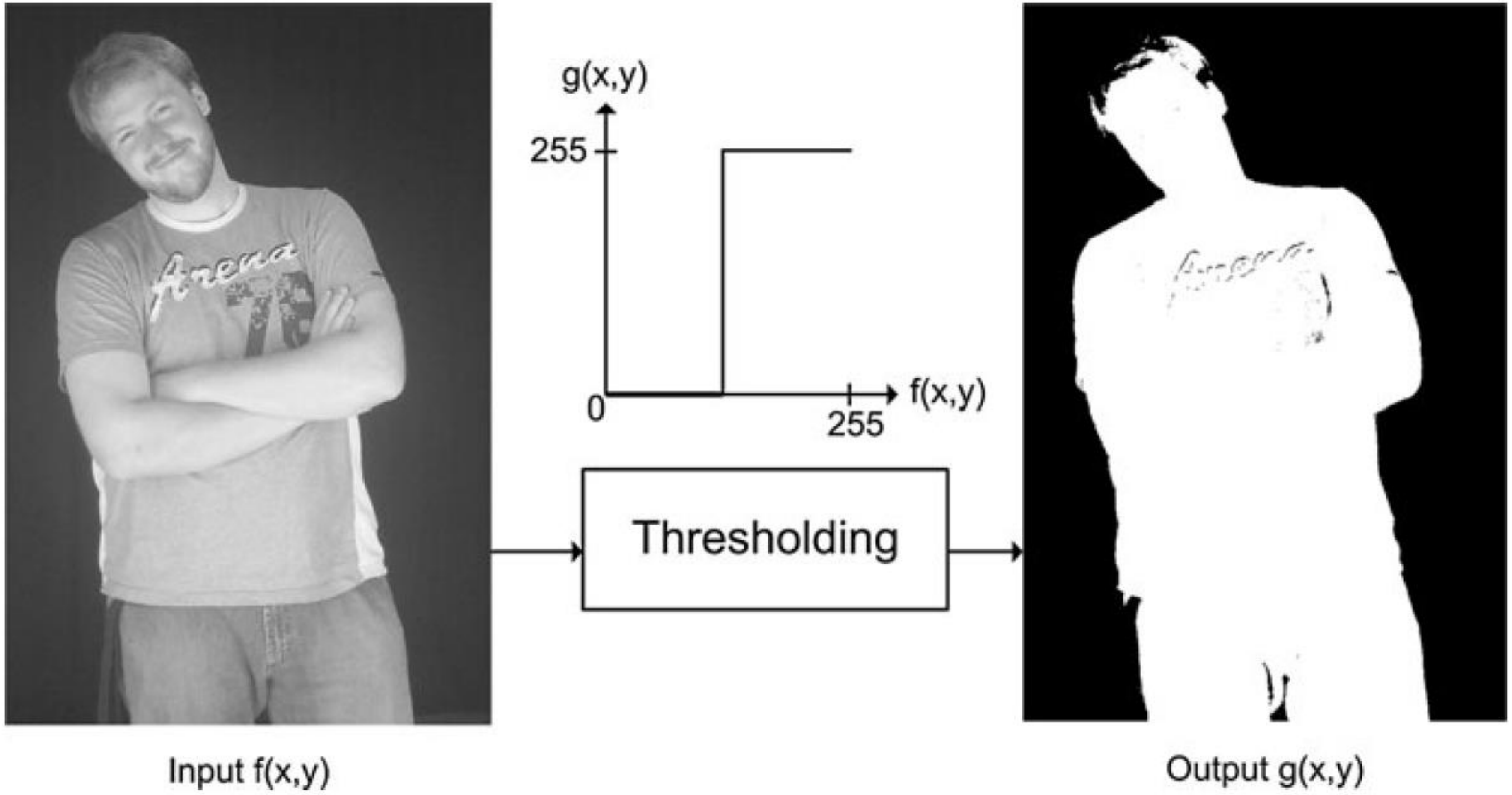
An example of cumulative histogram



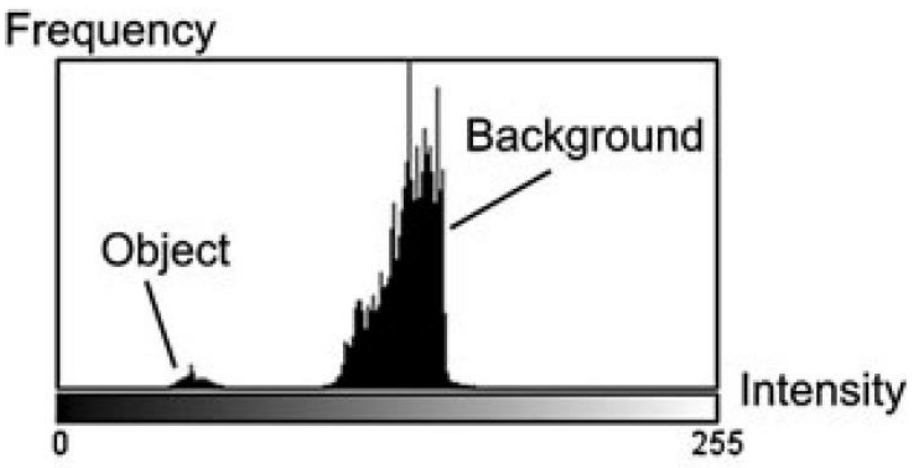
Application



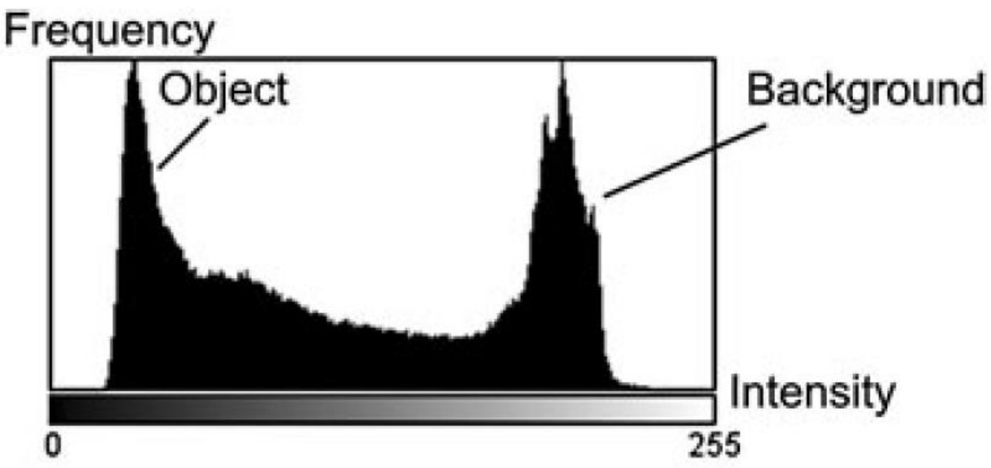
Thresholding



When does it work?

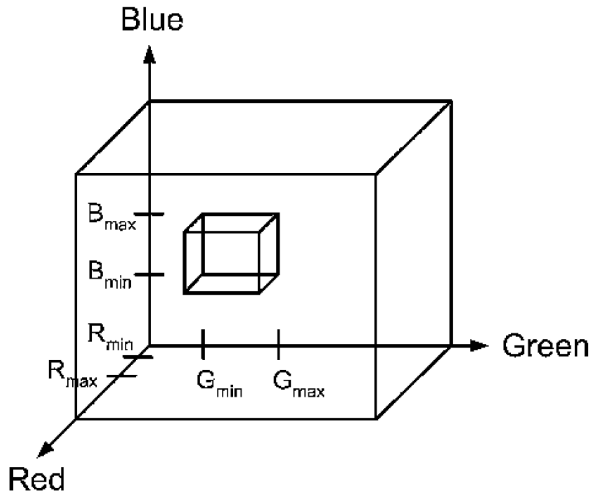


Ideal histogram



Problematic histogram

Color thresholding



If

$R > R_{\min}$ and $R < R_{\max}$ and

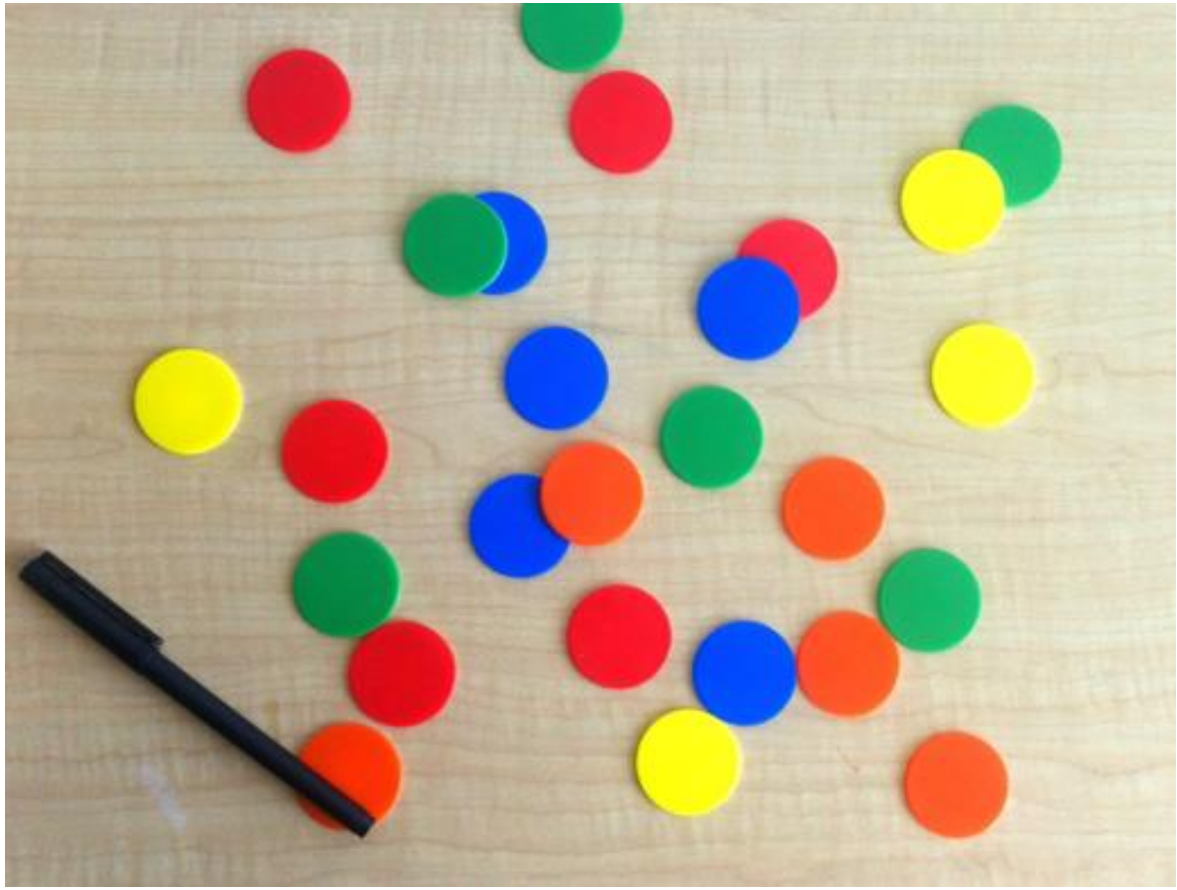
$G > G_{\min}$ and $G < G_{\max}$ and

$B > B_{\min}$ and $B < B_{\max}$

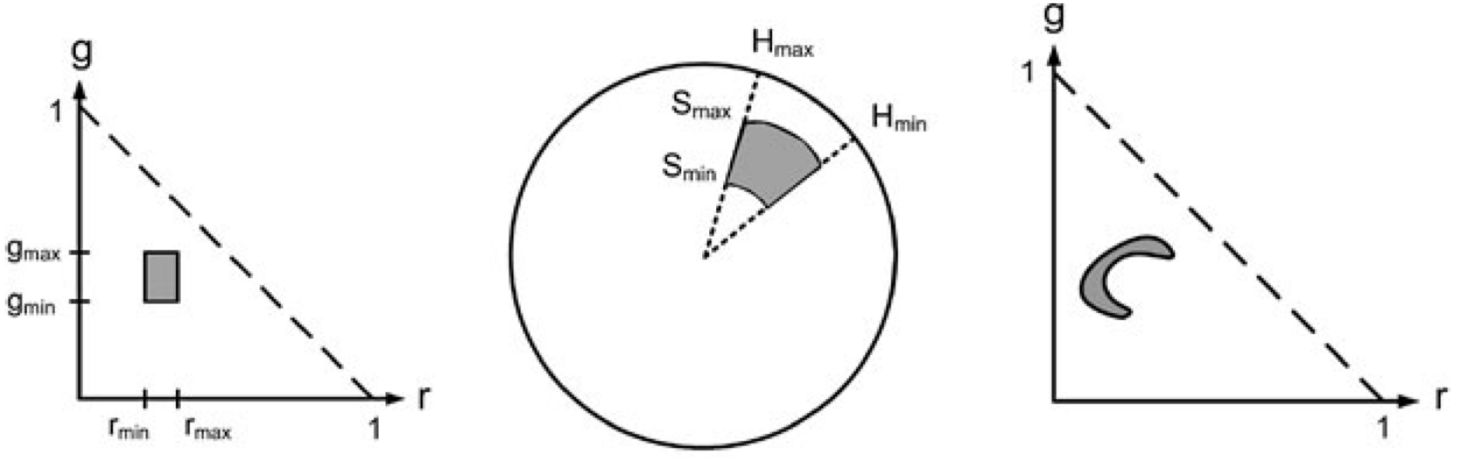
Then $g(x, y) = 255$

Else $g(x, y) = 0$

Application – how to find the blue chips?

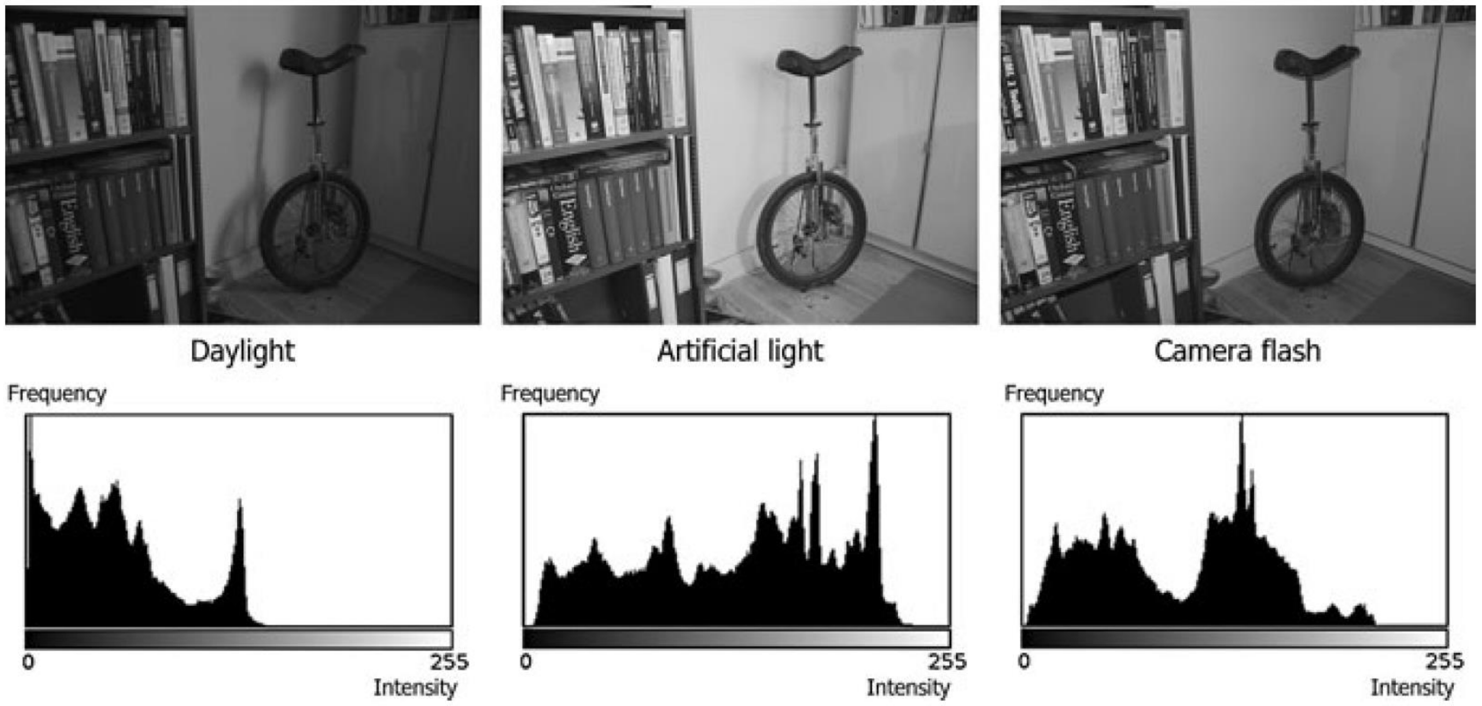


Problems



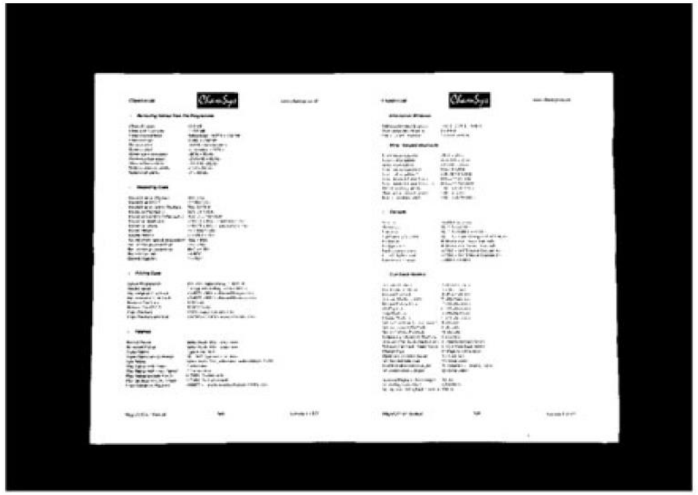
(a) (b) (c)

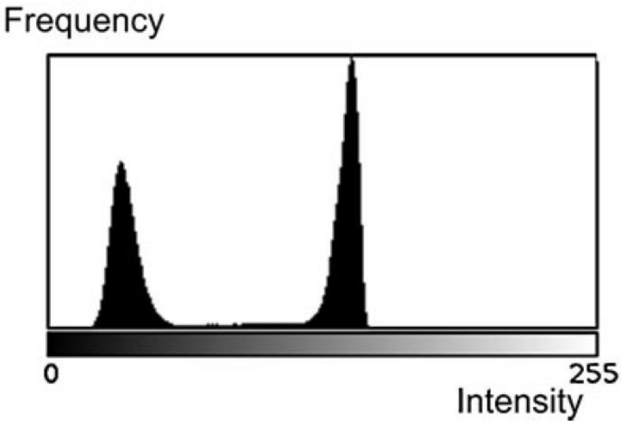
Thresholding in video

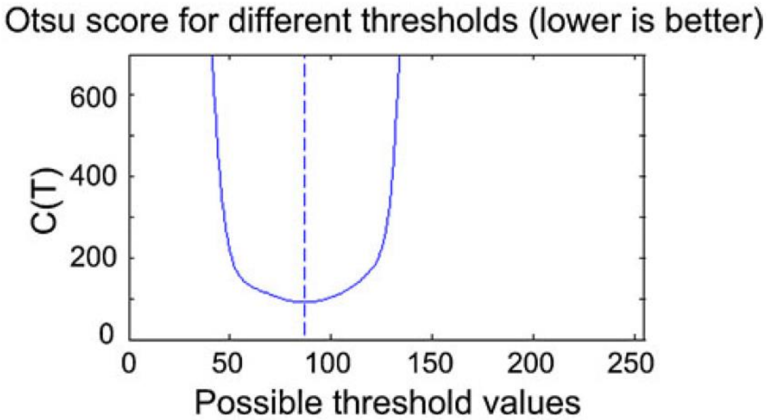


Automatic thresholding

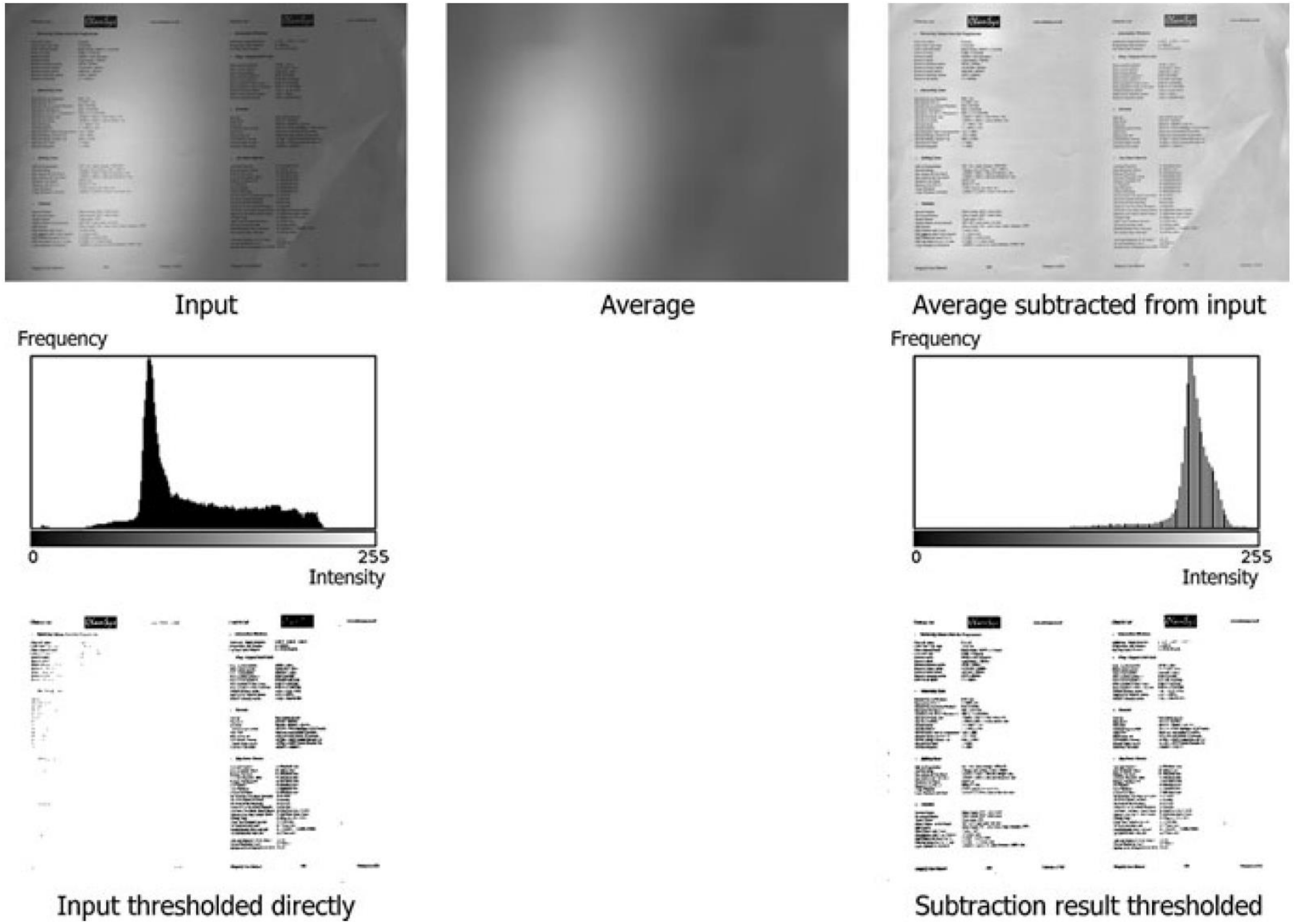
- Otsu score $$C(T) = M_1(T) \cdot \sigma_1^2(T) + M_2(T) \cdot \sigma_2^2(T)$$







Automatic thresholding (local)



Inverting an image

g(x,y)=255-f(x,y)

f(x,y)

g(x,y)

Alpha blending

$$g(x,y)=\alpha \cdot f_1(x,y)+(1-\alpha) \cdot f_2(x,y)$$