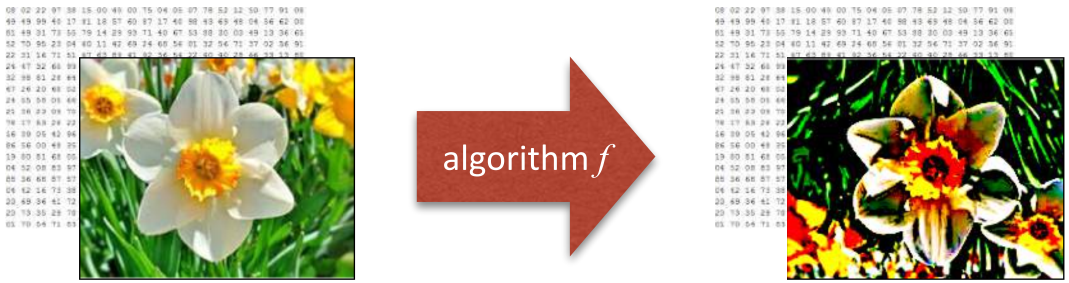
Histogram and point operations

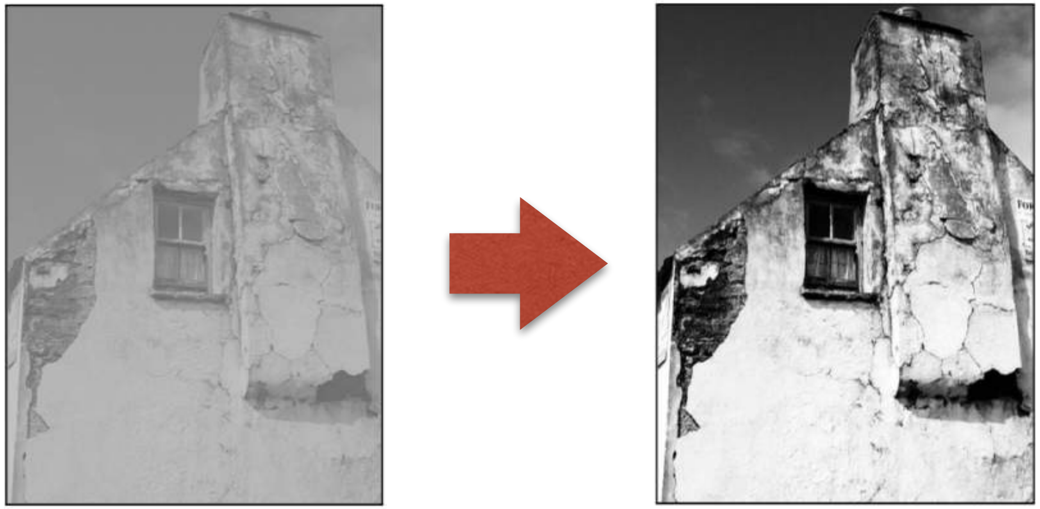
Recall... (1/3)

- Image processing is the study of any algorithm that takes an **image as input** and returns an **image as output**



I $\qquad$ $J = f(I)$

- Example: **contrast adjustment**



Input image : I $\qquad$ Output image : J

Recall... (2/3)

Images as functions

- We can think of the *intensity of an image* as a function of position (u, v)
- Let $\Omega \subset \mathbb{N}^2$ be the *image domain*. Then an image is a **discrete function**:

$$I : \Omega \rightarrow \mathbb{R}$$

Example

A simple image

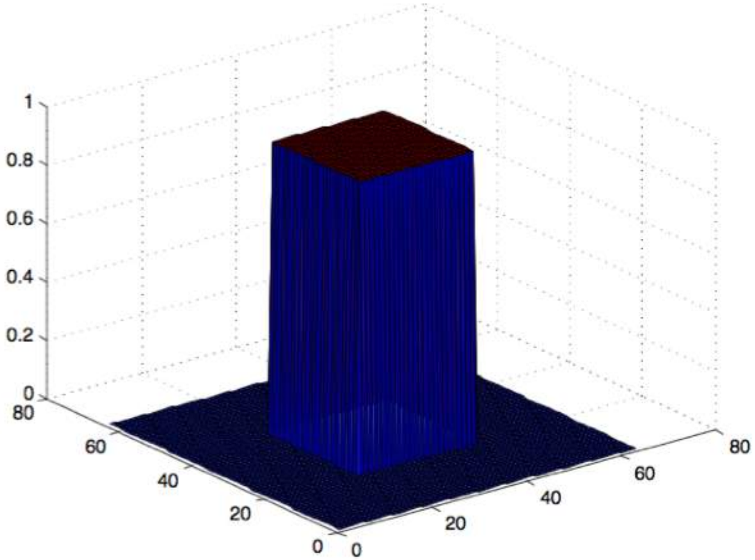
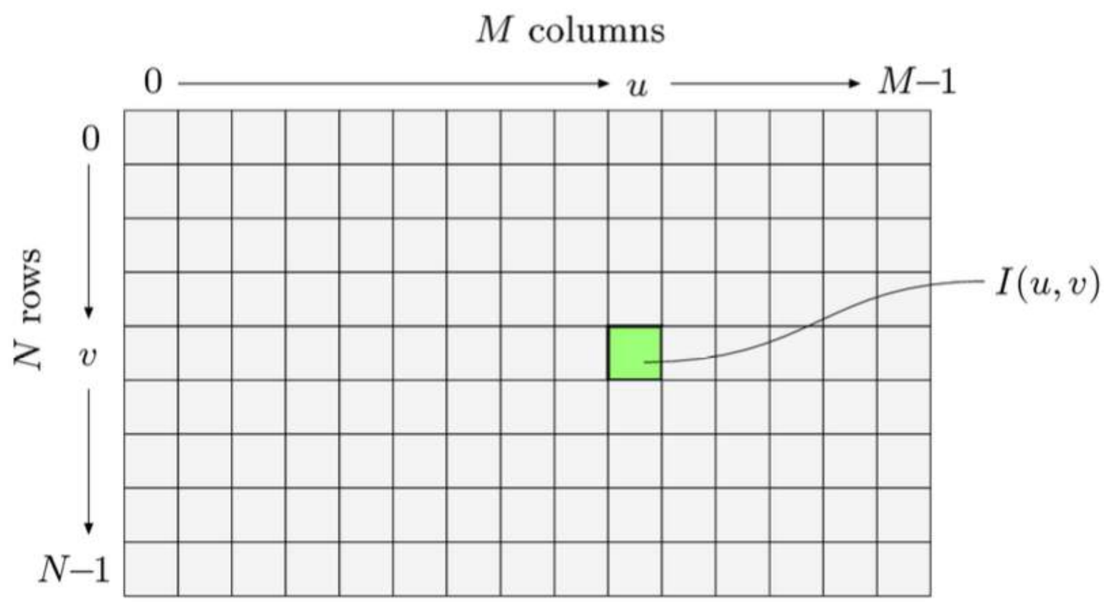


Image function as a height field

Representing an image

- The data structure for an image is simply a **2D array of values**:



- The values in the array can be any data type (8-bit, 16-bit…, signed/unsigned etc)

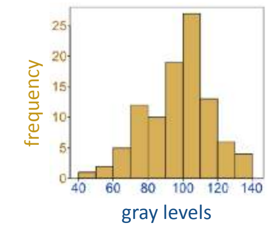
Note

- Here we work with **grayscale 2D images**…
- …but in medical imaging they can *have more dimensions and channels*!

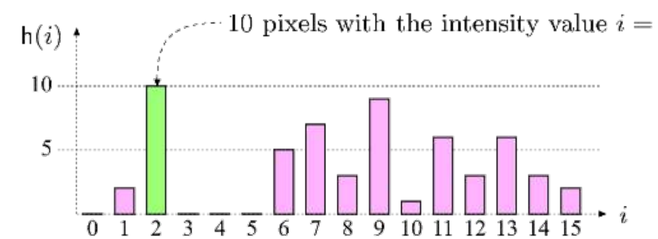
Histogram of an image (1/2)

- A **histogram** is a function $h(i)$ that gives the *frequency of each intensity* i that occur in an image
  - Given an image $I : \Omega \rightarrow [0 \ldots K-1]$, its histogram is the function:

$$h(i) = \text{card}\,\{\ (u,v) \mid I(u,v) = i\ \}$$

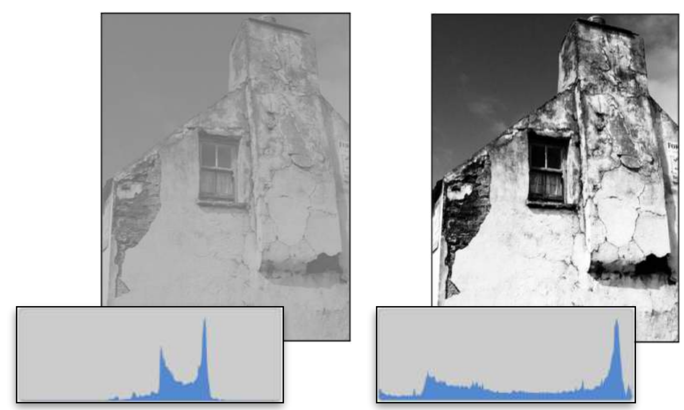


- In other words
  - $h(i)$ = number of pixels with intensity i



$\mathsf{h}(i)$	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

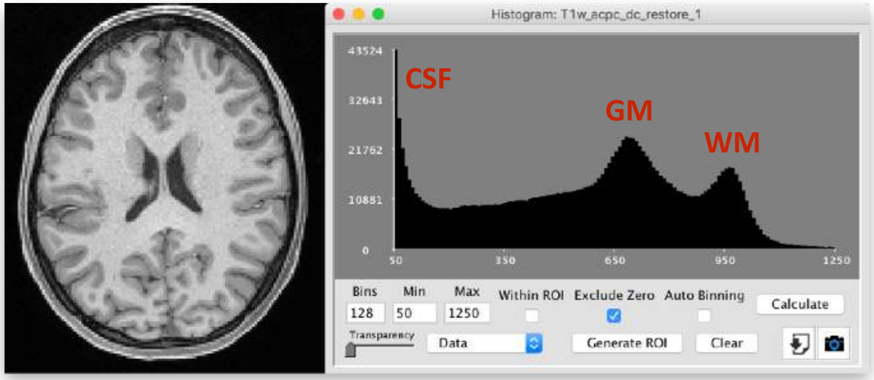
- Notes
  - *Low-contrast* image ➜ histogram is *narrow*
  - *High-contrast* image ➜ histogram is *spread out*
  - In general, *image processing* alters the histogram



Histogram of an image (2/2)

Probabilistic interpretation

- **Question**:
  - if we pick a pixel at random, what is the probability that the intensity of this voxel is equal to i?
- **Answer**:
  - P($I(u,v)=i$) = # pixels with value i / # pixels of the image
  - P($I(u,v)=i$) = $h(i)$ / MN



Generalization: a *binned histogram* gives the frequency of image intensities that fall into *small intervals* (or bins)

- Given an image $I : \Omega \rightarrow [0 \ldots K-1]$, its binned histogram is the function:

$$h(i) = \text{card}\left\{ (u,v) \mid a_i \leq I(u,v) < a_{i+1} \right\}$$

where $0 < a_0 < a_1 < \ldots < K$

- Typically, we choose equally spaced bins

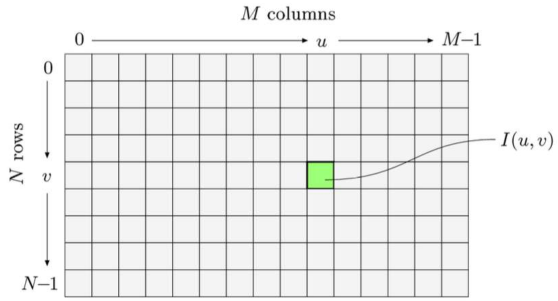
Definition of “point operation”

- A **point operation** on an image is an algorithm that *changes each pixel value* according to some function

$$J(u,v) \mapsto f\Big[I(u,v)\Big]$$

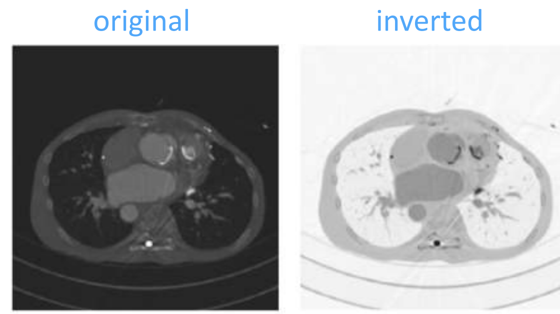
  - The function f <u>depends only on the pixel value</u>
  - It is *independent of the spatial location* (u, v)
  - The range of f determines the output datatype, e.g. uint16 or float32

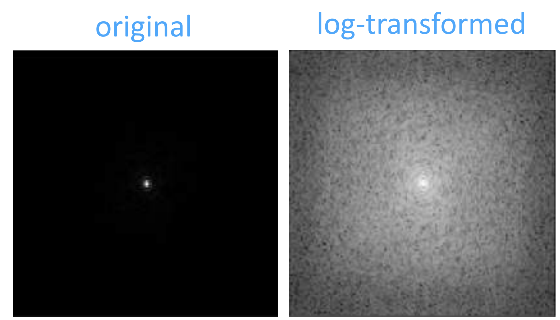
- Pseudocode
  - **Input**: image $I(u, v)$ defined on $[0 \ldots M\text{-}1] \times [0 \ldots N\text{-}1]$
  - **Output**: new image $J(u, v)$
  - *for* $v = 0 \ldots N\text{-}1$
    for $u = 0 \ldots M\text{-}1$
    set $J(u, v) = f[\, I(u, v)\,]$



- Any function ca be used, e.g. $f(x)=x^2$, but *not all are useful*

- Most common
  - Addition and multiplication
    - $f(p) = p + k$
    - $f(p) = p \cdot x$
  - Inverse
    - $f(p) = L - p$
  - Log transform
    - $f(p) = \log(1+p)$
  - Gamma correction
    - $f(p) = p^\gamma$





- Notes
  - What happens to their *histograms*?
  - Beware of *output data type* (overflow)

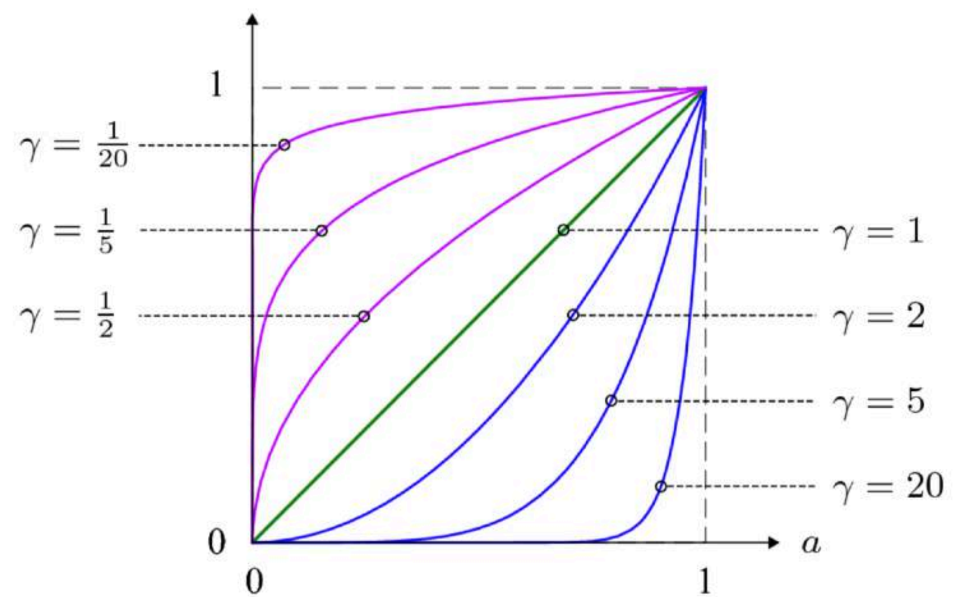
Basic examples

Gamma correction

- The **human perception** of brightness follows an approximate *power function*

$$f(a) = a^{\gamma}$$

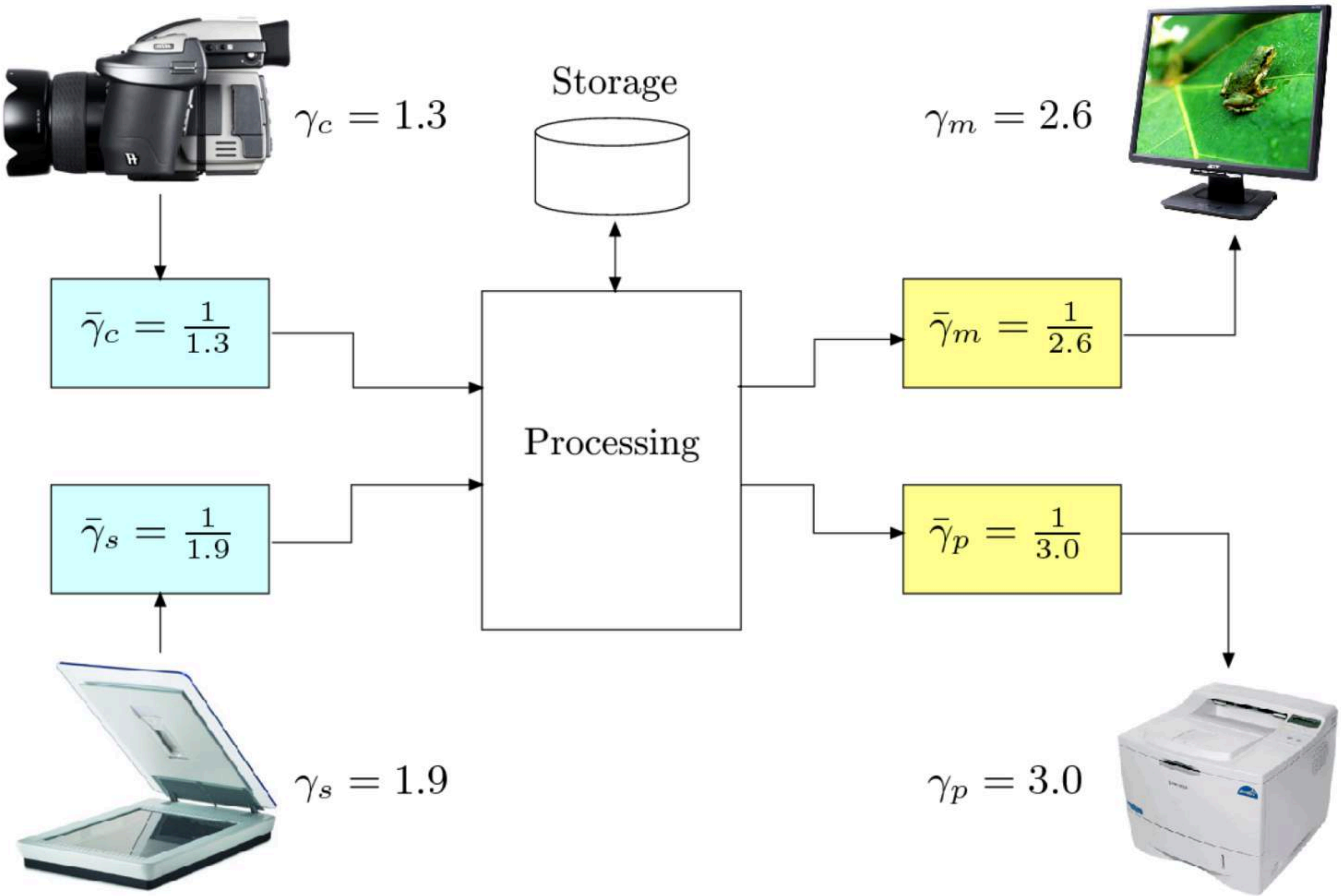
- **Greater sensitivity** to differences between *darker tones* (than between lighter ones)
- When forming an image, **camera sensors** convert light into an electrical signal
  - NB: different sensors may have different responses to light intensity and produce different signals
- **Display devices** (monitors, printers) turn images into a physical representation (light, ink)



Q: how do we make sure there is **consistency**?

A: hardware manufacturers **specify the gamma value** to properly *record* or *reproduce* the colors

Basic examples (3/3)

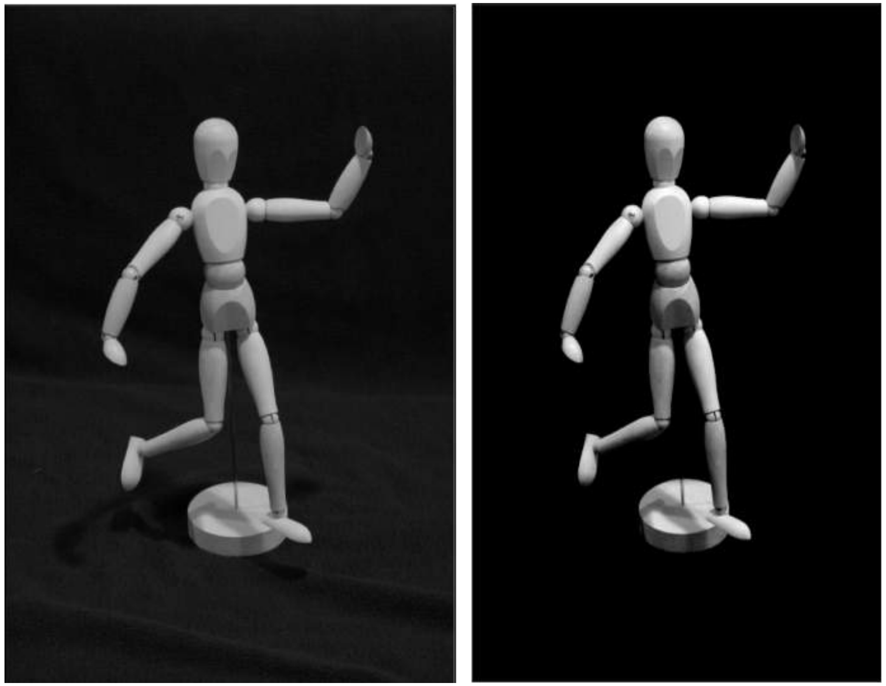


Useful functions (1/2)

Clamping (or clipping)

- *Limit intensities* to a given interval $[a, b]$

$$f(p) = \begin{cases} a & \text{if } p < a \\ p & \text{if } a \leq p \leq b \\ b & \text{if } p > b \end{cases}$$



Windowing

- *Clamping* followed by *intensity stretching* to fill the full possible range $[0, M]$

$$f(p) = \begin{cases} 0 & \text{if } p < a \\ M \times \frac{p-a}{b-a} & \text{if } a \leq p \leq b \\ M & \text{if } p > b \end{cases}$$

Useful functions (2/2)

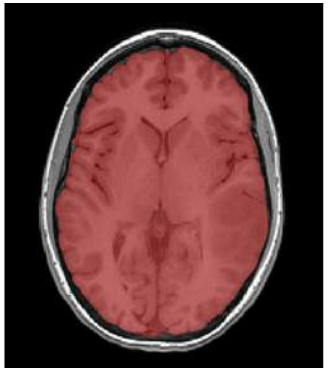
Thresholding

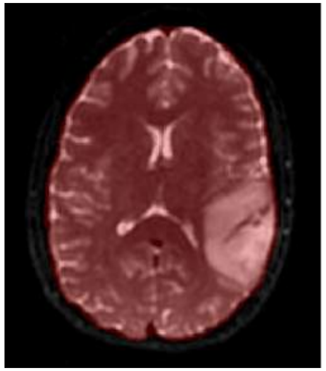
- Also called *image binarization*

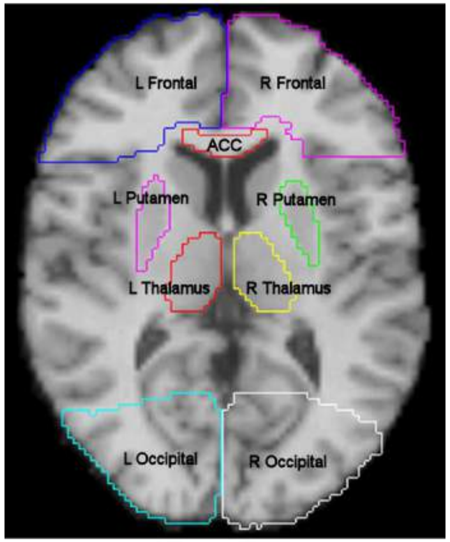
$$f(p) = \begin{cases} 0 & \text{if } p \leq a \\ 1 & \text{if } p > a \end{cases}$$

Notes

- Despite their simplicity, **binary images** are widely used in medical image processing
- <u>Examples</u>
  - *Constrain the processing* to a given portion of the image, e.g brain
  - *Regions-of-interest* (ROI) analysis



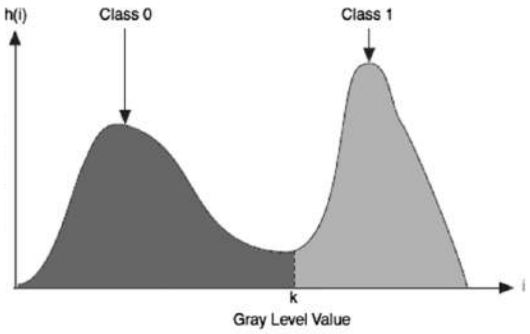




Difficult to find the **proper threshold**

- Usually, there are **not only two regions**, e.g. foreground and background
- We will see **advanced segmentation tools** for the more general case

OTSU's method



- *Very basic* algorithm to **automatically binarize an image**
- Assumes that the image contains *exactly two regions* (i.e. bimodal histogram)
  - Actually, extensions exist for multiple regions
- Searches for the threshold that **minimizes the intra-class variance**:

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$$

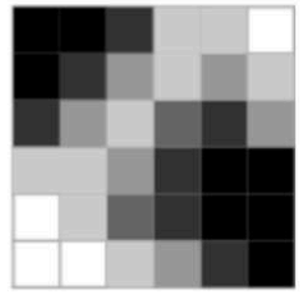
$$\omega_0(t) = \sum_{i=0}^{t-1} p(i)$$

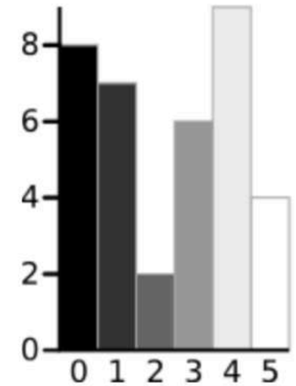
$$\omega_1(t) = \sum_{i=t}^{L-1} p(i)$$

  - ω_0 and ω_1 are the *probabilities of the two classes* separated by a *threshold t*
  - σ^2_0 and σ^2_1 are *variances* of these two classes
- **NB**: iterates through all the possible threshold values

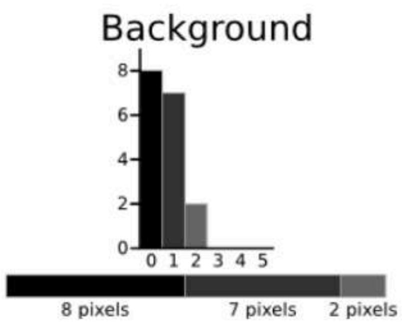
Toy example

- 6x6 image
- 6 gray levels





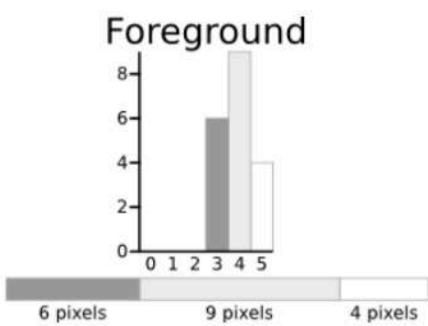
Calculations for the case t=3



$$\text{Weight} \quad W_b = \frac{8+7+2}{36} = 0.4722$$

$$\text{Mean} \quad \mu_b = \frac{(0\times 8)+(1\times 7)+(2\times 2)}{17} = 0.6471$$

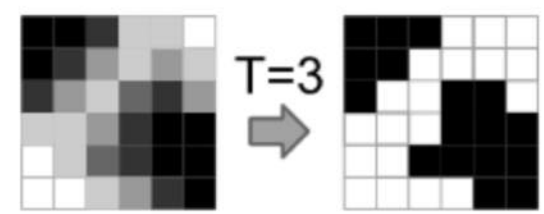
$$\text{Variance} \quad \sigma_b^2 = \frac{((0-0.6471)^2\times 8)+((1-0.6471)^2\times 7)+((2-0.6471)^2\times 2)}{17} = \frac{(0.4187\times 8)+(0.1246\times 7)+(1.8304\times 2)}{17} = 0.4637$$



$$\text{Weight} \quad W_f = \frac{6+9+4}{36} = 0.5278$$

$$\text{Mean} \quad \mu_f = \frac{(3\times 6)+(4\times 9)+(5\times 4)}{19} = 3.8947$$

$$\text{Variance} \quad \sigma_f^2 = \frac{((3-3.8947)^2\times 6)+((4-3.8947)^2\times 9)+((5-3.8947)^2\times 4)}{19} = \frac{(4.8033\times 6)+(0.0997\times 9)+(4.8864\times 4)}{19} = 0.5152$$

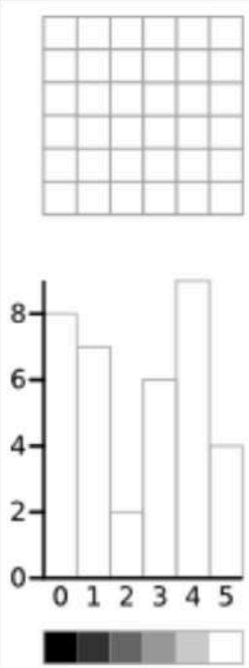
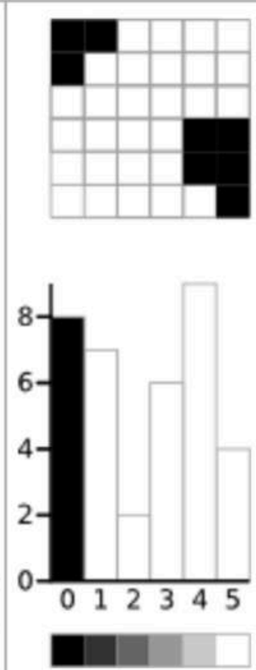
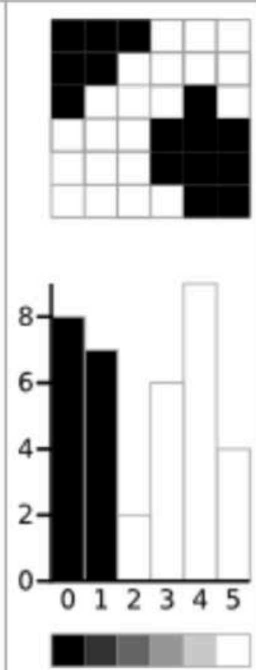
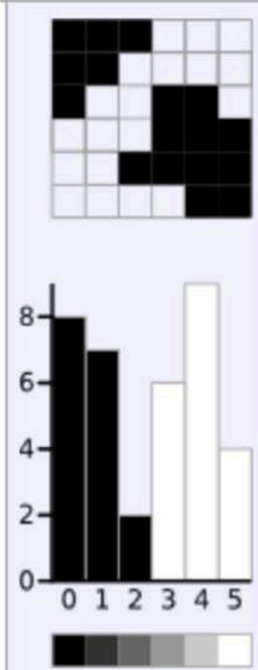
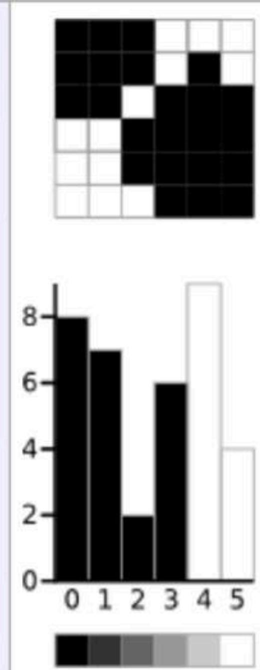
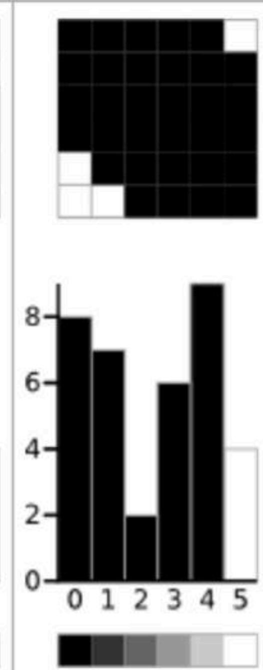


Within-class variance

$$\boxed{\sigma_W^2 = W_b\,\sigma_b^2 + W_f\,\sigma_f^2 = 0.4722 * 0.4637 + 0.5278 * 0.5152 = 0.4909}$$

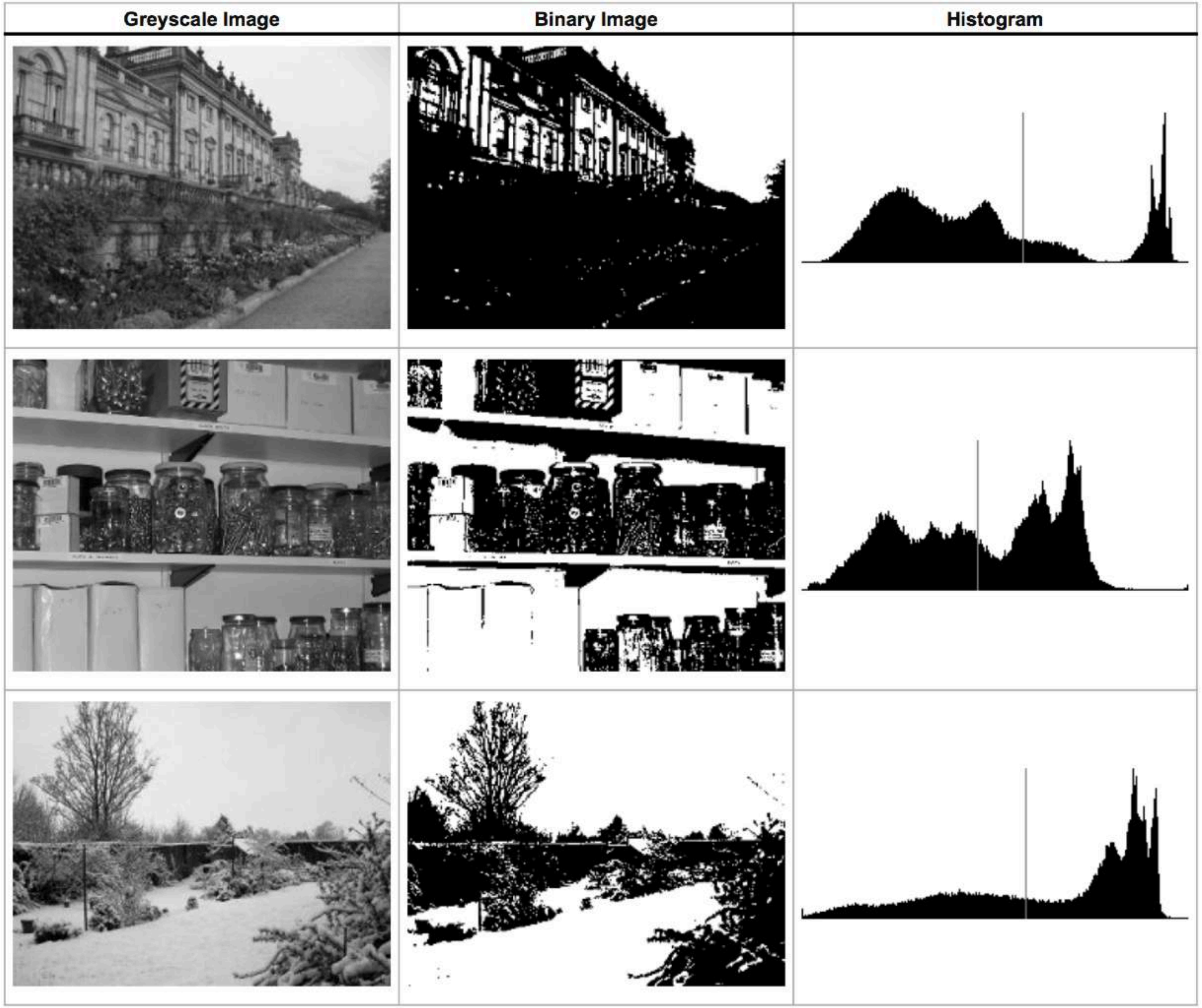
Automatic thresholding

- Iterate over all threshold values T

Threshold	T=0	T=1	T=2	T=3	T=4	T=5
						
Weight, Background	$W_b = 0$	$W_b = 0.222$	$W_b = 0.4167$	$W_b = 0.4722$	$W_b = 0.6389$	$W_b = 0.8889$
Mean, Background	$M_b = 0$	$M_b = 0$	$M_b = 0.4667$	$M_b = 0.6471$	$M_b = 1.2609$	$M_b = 2.0313$
Variance, Background	$\sigma^2_b = 0$	$\sigma^2_b = 0$	$\sigma^2_b = 0.2489$	$\sigma^2_b = 0.4637$	$\sigma^2_b = 1.4102$	$\sigma^2_b = 2.5303$
Weight, Foreground	$W_f = 1$	$W_f = 0.7778$	$W_f = 0.5833$	$W_f = 0.5278$	$W_f = 0.3611$	$W_f = 0.1111$
Mean, Foreground	$M_f = 2.3611$	$M_f = 3.0357$	$M_f = 3.7143$	$M_f = 3.8947$	$M_f = 4.3077$	$M_f = 5.000$
Variance, Foreground	$\sigma^2_f = 3.1196$	$\sigma^2_f = 1.9639$	$\sigma^2_f = 0.7755$	$\sigma^2_f = 0.5152$	$\sigma^2_f = 0.2130$	$\sigma^2_f = 0$
Within Class Variance	$\sigma^2_W = 3.1196$	$\sigma^2_W = 1.5268$	$\sigma^2_W = 0.5561$	$\boldsymbol{\sigma^2_W = 0.4909}$	$\sigma^2_W = 0.9779$	$\sigma^2_W = 2.2491$

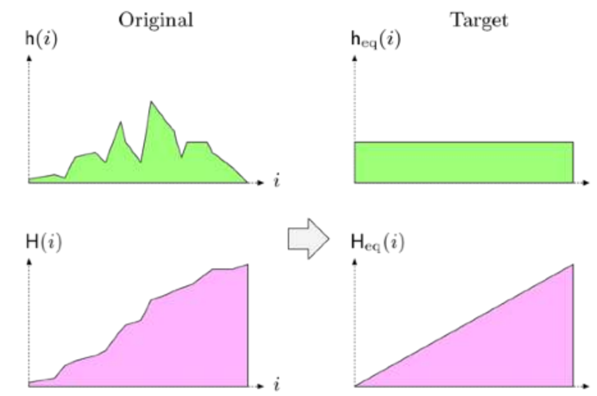
Automatic thresholding (4/4)

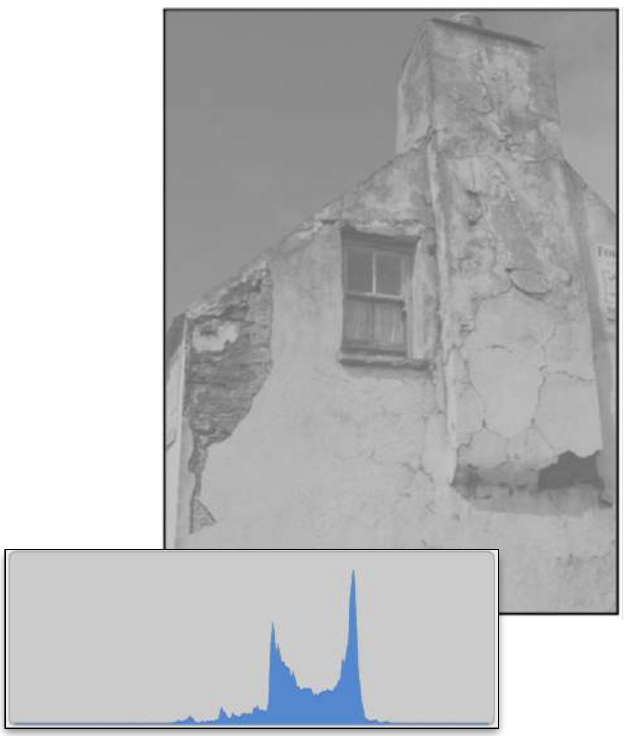
- Examples

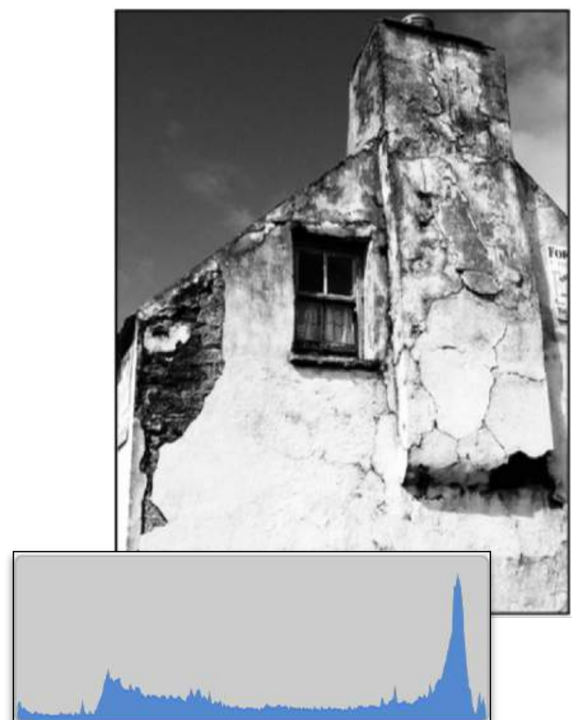


Automatic contrast adjustment (1/2)

- **Histogram equalization** aims at improving the contrast by *rescaling the histogram*
  - *Low-contrast images* use only portion of the available gray levels, i.e. narrow histogram
  - After *equalization*, histogram is *spread out*
  - The intensities now range over *all possible gray levels*

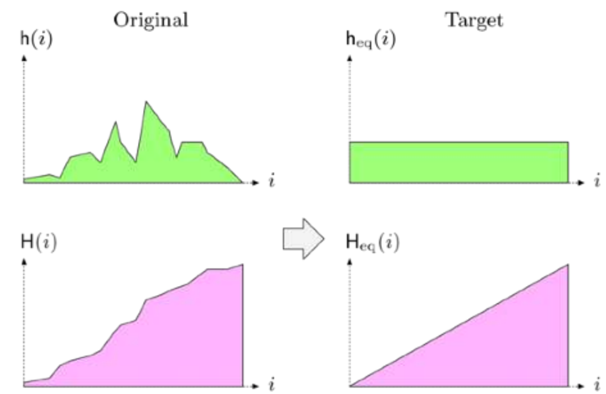






Automatic contrast adjustment (1/2)

- **Histogram equalization** aims at improving the contrast by *rescaling the histogram*
  - *Low-contrast images* use only portion of the available gray levels, i.e. narrow histogram
  - After *equalization*, histogram is *spread out*
  - The intensities now range over *all possible gray levels*



- Based on the *cumulative distribution function* of $h(i)$ (also called **cumulative histogram**)
  - $H(i)$ = number of pixels with an *intensity less than or equal* to i

$$H(i) = \sum_{j=0}^{i} h(j)$$

Automatic contrast adjustment (2/2)

Procedure

(1) Compute the *histogram* $h(i)$ of the image

(2) *Normalize* $h(i)$ s.t. its range is $[0,\, 1]$ (i.e. divide by the total number of pixels)

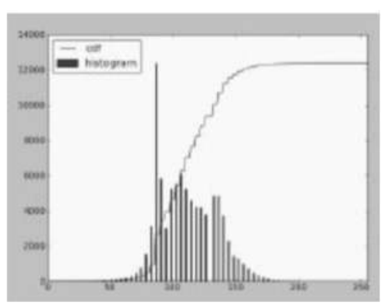
(3) Calculate the *cumulative histogram* $H(i)$

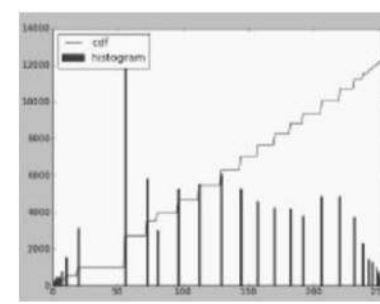
(4) Apply the following *point operation function* to every pixel p:

$$f(p) = \operatorname{round}\left(\frac{H(p) - H_{\min}}{1 - H_{\min}} \cdot (L-1)\right)$$

where L is the number of gray levels of the image (i.e. [0, L-1] range)

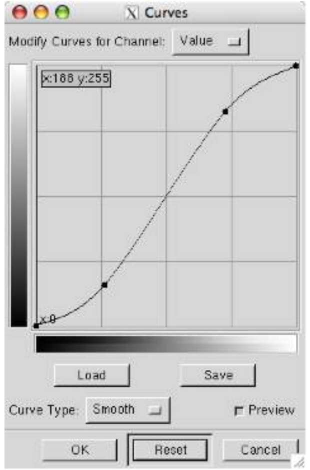
Example





"Curves" operation

- Most *software packages* allow us to manually alter histograms by adjusting the contrast with a **continuous curve**



- **Continuous** point-operation functions
  - Let's assume for simplicity:
    - That the image has *continuous pixel type* (floating point)
    - The *intensity range* is in the interval $[0, 1]$, i.e. $I : \Omega \rightarrow [0, 1]$
  - A *continuous point-operation* is any function

$$f : [0, 1] \rightarrow [0, 1]$$

- The contrast is changed as…
  - slope = 1 ➜ no contrast change
  - slope < 1 ➜ contrast is decreased
    (wide range of values mapped *to smaller range* of values)
  - slope > 1 ➜ contrast is increased
    (stretches small range of values *to larger range* of values)

High Dynamic Range (HDR) imaging (1/8)

- **FACT**: sensors in digital cameras capture *much less dynamic range* than the human eye
  - **Dynamic Range** = *ratio* between *maximum* and *minimum* measurable light intensities

- Examples

your eyes

- **FACT**: sensors in digital cameras capture *much less dynamic range* than the human eye
  - **Dynamic Range** = *ratio* between *maximum* and *minimum* measurable light intensities

- Examples

your camera

- **FACT**: sensors in digital cameras capture *much less dynamic range* than the human eye
  - **Dynamic Range** = *ratio* between *maximum* and *minimum* measurable light intensities

- Examples

your eyes

High Dynamic Range (HDR) imaging (1/8)

- **FACT**: sensors in digital cameras capture *much less dynamic range* than the human eye
  - **Dynamic Range** = *ratio* between *maximum* and *minimum* measurable light intensities
- Examples

your camera

High Dynamic Range (HDR) imaging (2/8)

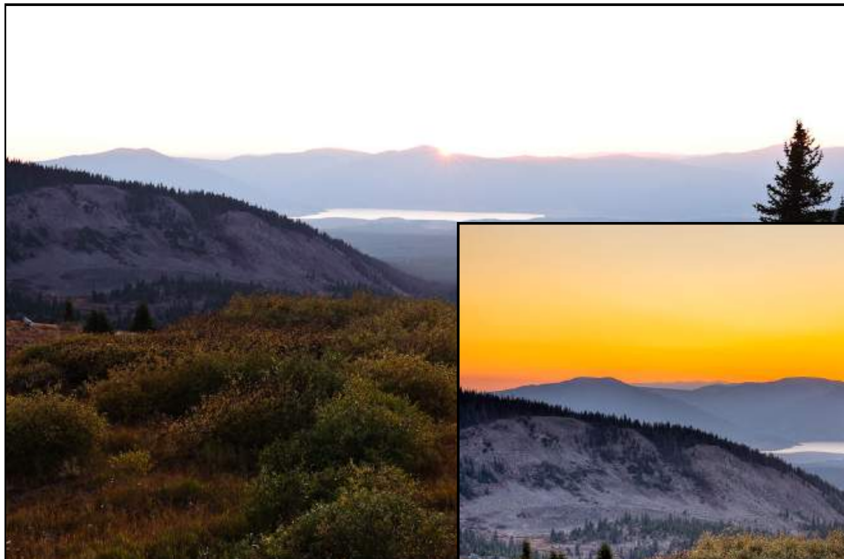
- What is the problem?

Try to **correctly expose the ground**,
but end up completely blowing out the sky

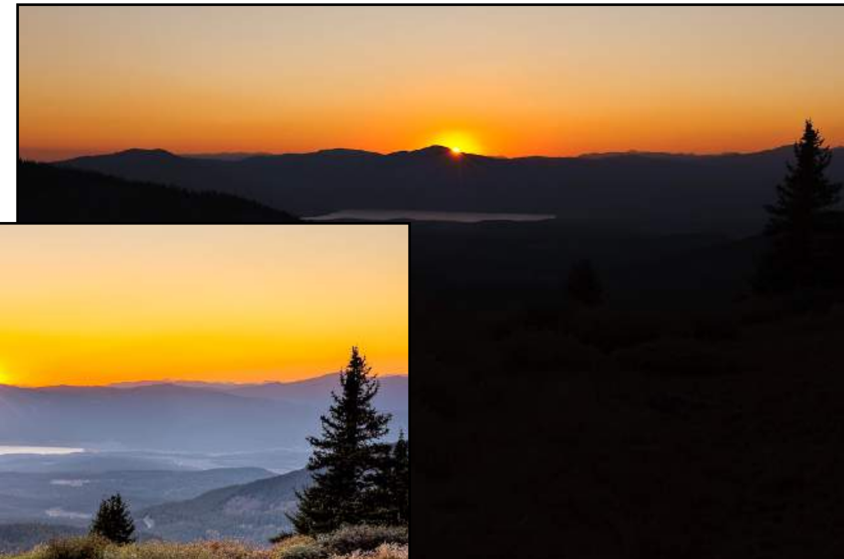
Try to **correctly expose the sky**,
but the ground becomes too dark to see any detail

- Clearly, camera is unable to capture such a **large dynamic range**

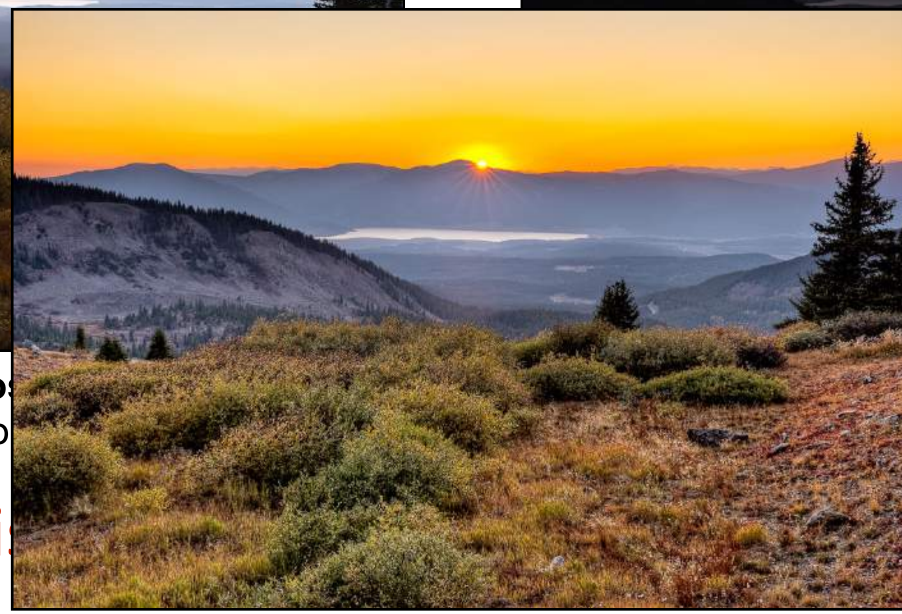
- What is the problem?



Try to **correctly expo**
but end up completely b



y expose the sky,
s too dark to see any detail



- Clearly, camera i **ge dynamic range**
- HDR imaging is a way to **capture all the dynamic range of a scene** by combining multiple exposures
  - This resembles **what our eyes do**: they adapt the "exposure" when looking at different portions of the scene having different light intensities

High Dynamic Range (HDR) imaging

- There are **many different techniques** for HDR photography
  - e.g. *Tone Mapping*, *Digital Blending*, *Dynamic Range Increase*, *Luminance Masking*…

- The **basic idea** is:
  - Capture the same scene with <u>different exposures</u>
    - **Normal**: standard exposure
    - **Under-exposed**: to get details in the *lightest* areas
    - **Over-exposed**: to get details in the *darkest* areas

EV: **-2** EV: **0** EV: **+2**

  - <u>Combine them</u> together into one image to obtain contrast in each area of the image

- This way, **all areas** of the image will be **properly exposed**!

Examples of different functions

EV: +2

EV: 0

EV: -2

Natural

High Dynamic Range (HDR) imaging (4/8)

- Examples of different functions

EV: +2

EV: 0

EV: -2

Balanced

- Examples of different functions

EV: +2

EV: 0

EV: -2

Painterly

- Examples of different functions

EV: +2

EV: 0

EV: -2

Surreal

High Dynamic Range (HDR) imaging (5/8)

- HDR photography has become a **photographic phenomenon**
- **Most cameras** (also phones) offer the possibility to shoot in HDR
  - This function is usually called ***A**uto **E**xposure **B**racketing* (AEB)
  - Automatically *acquire multiple exposures* with a single click
  - Multiple images are saved

- *Moving objects* create "**ghosts**"
  - **NB**: use the tripod!

- Same love it, some hate it
  - It relies heavily on *post-processing*
  - So, the output image can really be *whatever you want* it to be!!!

High Dynamic Range (HDR) imaging (6/8)

- Some **good** examples

High Dynamic Range (HDR) imaging (6/8)

- Some **good** examples

High Dynamic Range (HDR) imaging (6/8)

- Some **good** examples

- Some **good** examples

High Dynamic Range (HDR) imaging (7/8)

- Some **extreme** examples

High Dynamic Range (HDR) imaging (7/8)

- Some **extreme** examples

- Some **extreme** examples

- Some **extreme** examples

High Dynamic Range (HDR) imaging (8/8)

- Definitely, **too much** post processing…

High Dynamic Range (HDR) imaging (8/8)

- Definitely, **too much** post processing…

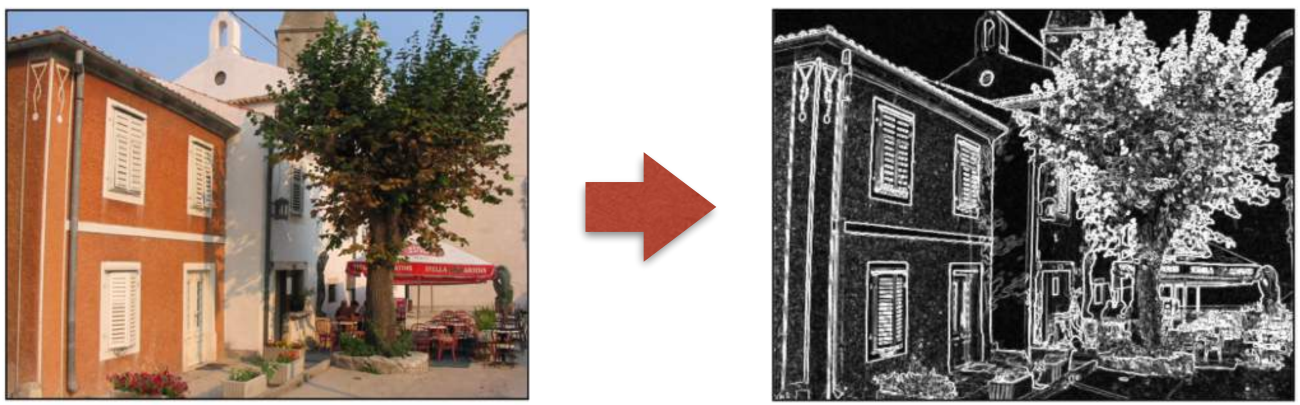
- Definitely, **too much** post processing…

High Dynamic Range (HDR) imaging (8/8)

- Definitely, **too much** post processing…

Limitations of point operations

- Main limitations
  - They don’t know *where they are* in an image
  - They don’t know anything about their *neighbors*
- Most image features (e.g. edges) involve a **spatial neighborhood** of pixels



Requires derivatives (i.e. spatial information)

- If we want to enhance these features we need to **go beyond point operations**