Basic image processing

Yoni Chechik

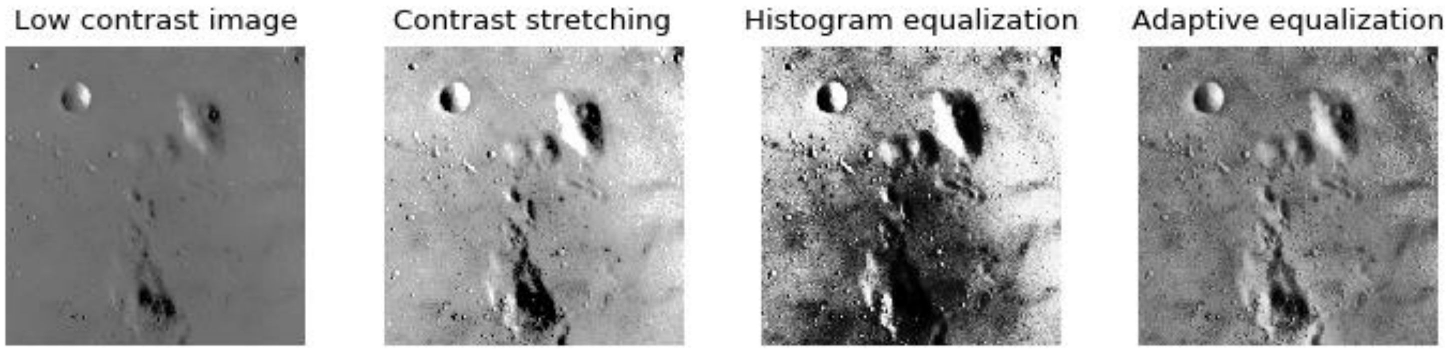
www.AlisMath.com

References

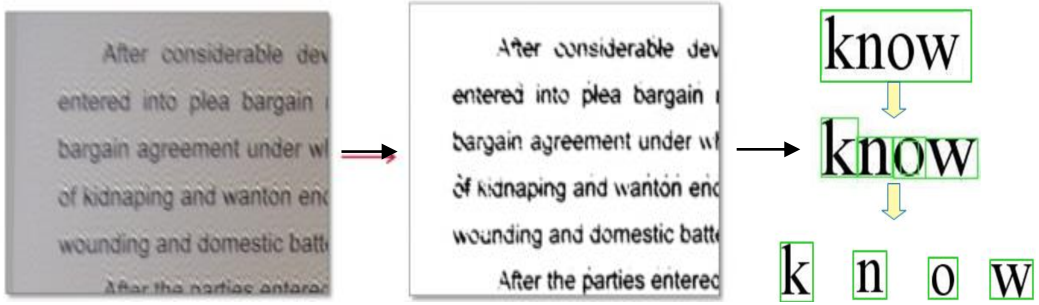
- http://szeliski.org/Book/
- http://www.cs.cornell.edu/courses/cs5670/2019sp/lectures/lectures.html
- http://www.cs.cmu.edu/~16385/

Some motivation

Art
(Photoshop color grading)



Science and space
(image enhancement)



Robotics
(OCR – optical character recognition)

Agriculture
(color ripeness detection)

contents

- **Image representation**
- Pixel-wise operations
- Histogram equalization
- Template matching
- Morphology operators
- Connected components
- Color space

Image representation

- We can think of an image as a 3d matrix of discrete RGB values.
- The values mark the intensity of each color channel and are usually of type uint8 = $\{0, \ldots, 255\}$.

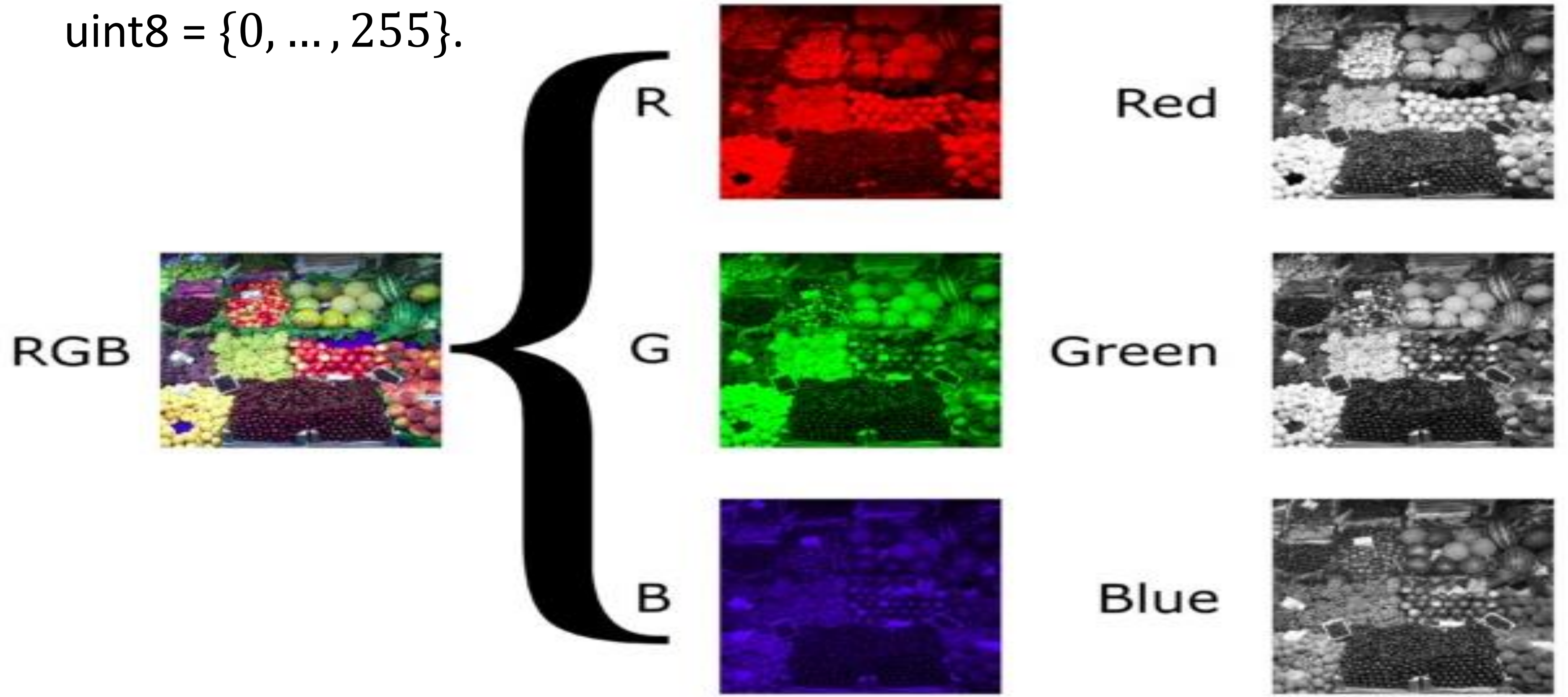
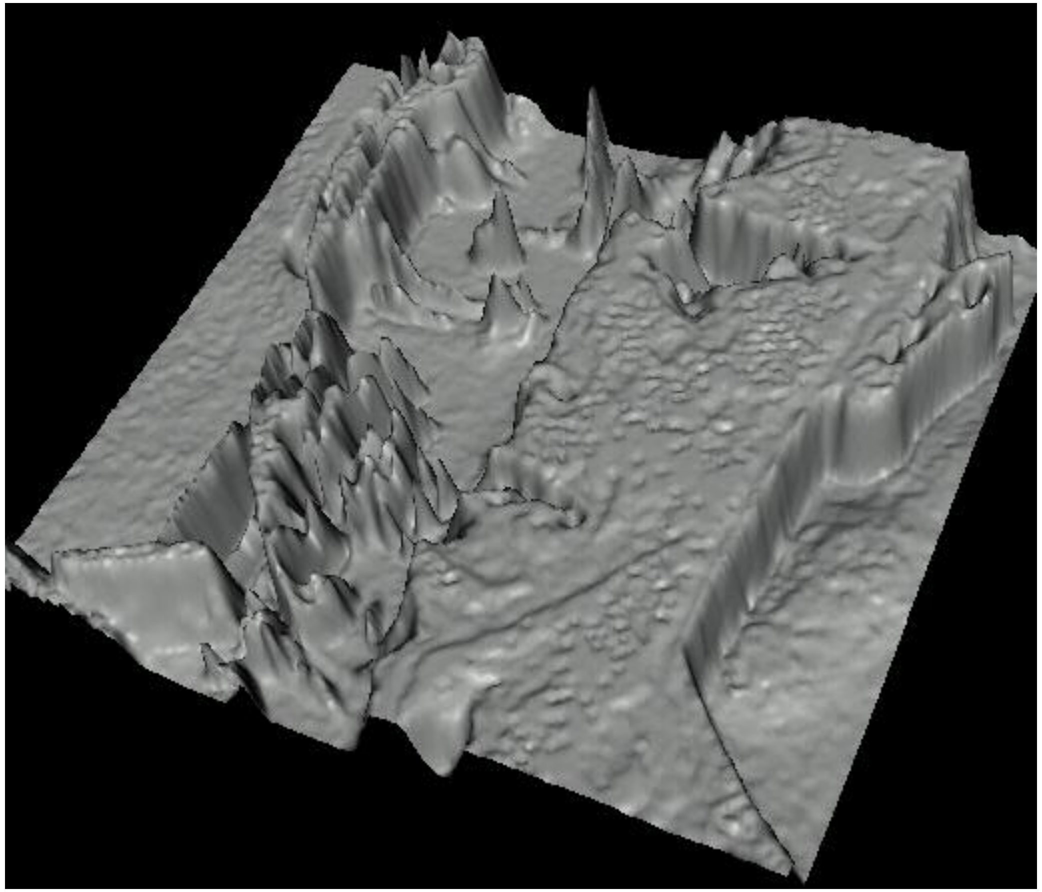


Image representation

- We can also think of an image as a function $f(x, y)$.



contents

- Image representation
- **Pixel-wise operations**
- Histogram equalization
- Template matching
- Morphology operators
- Connected components
- Color space

Pixel-wise operators

- Pixel-wise operators, or point operators, are defined as such that each output pixel's value depends on only the corresponding input pixel value.

Pixel-wise operators

original

x

darken

lower contrast

Gamma compression

invert

lighten

raise contrast

Gamma expansion

Pixel-wise operators

original

x

darken

lower contrast

Gamma compression

invert

$255 - x$

lighten

raise contrast

Gamma expansion

Pixel-wise operators

original

x

darken

$x - 128$

lower contrast

Gamma compression

invert

$255 - x$

lighten

raise contrast

Gamma expansion

Pixel-wise operators

original

x

darken

$x - 128$

lower contrast

Gamma compression

invert

$255 - x$

lighten

$x + 128$

raise contrast

Gamma expansion

Pixel-wise operators

original

x

darken

$x - 128$

lower contrast

$\frac{x}{2}$

Gamma compression

invert

$255 - x$

lighten

$x + 128$

raise contrast

Gamma expansion

Pixel-wise operators

original

x

darken

$x - 128$

lower contrast

$\frac{x}{2}$

Gamma compression

invert

$255 - x$

lighten

$x + 128$

raise contrast

$x \times 2$

Gamma expansion

Contrast

- **Contrast** in visual perception is the difference in appearance of two or more parts of a seen field.
- The human visual system is more sensitive to contrast than absolute luminance;
- **Contrast ratio**, or **dynamic range**, is the ratio between the largest and smallest values of the image or:

$$CR = \frac{V_{max}}{V_{min} + \epsilon}$$

Contrast

- Example of calculating contrast ratio in determining website accessibility:
  - https://contrast-ratio.com/#%23000000-on-white
  - https://www.accessibility-developer-guide.com/knowledge/colours-and-contrast/how-to-calculate/

Pixel-wise operators

original

x

darken

$x - 128$

lower contrast

$\frac{x}{2}$

Gamma compression

invert

$255 - x$

lighten

$x + 128$

raise contrast

$x \times 2$

Gamma expansion

Pixel-wise operators

original

x

darken

$x - 128$

lower contrast

$\frac{x}{2}$

Gamma compression

$\left(\frac{x}{255}\right)^{1/3} \times 255$

invert

$255 - x$

lighten

$x + 128$

raise contrast

$x \times 2$

Gamma expansion

Pixel-wise operators

original

$$x$$

darken

$$x - 128$$

lower contrast

$$\frac{x}{2}$$

Gamma compression

$$\left(\frac{x}{255}\right)^{1/3} \times 255$$

invert

$$255 - x$$

lighten

$$x + 128$$

raise contrast

$$x \times 2$$

Gamma expansion

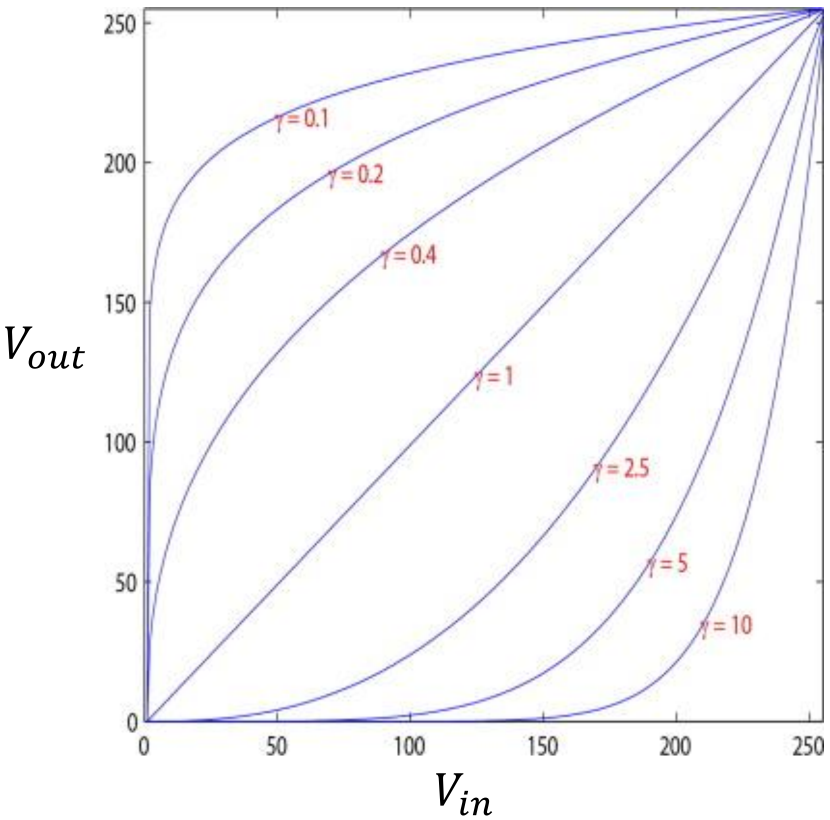
$$\left(\frac{x}{255}\right)^{2} \times 255$$

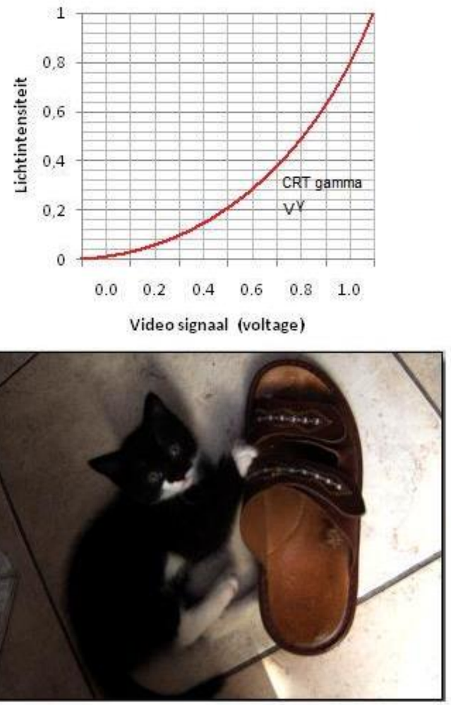
Gamma correction

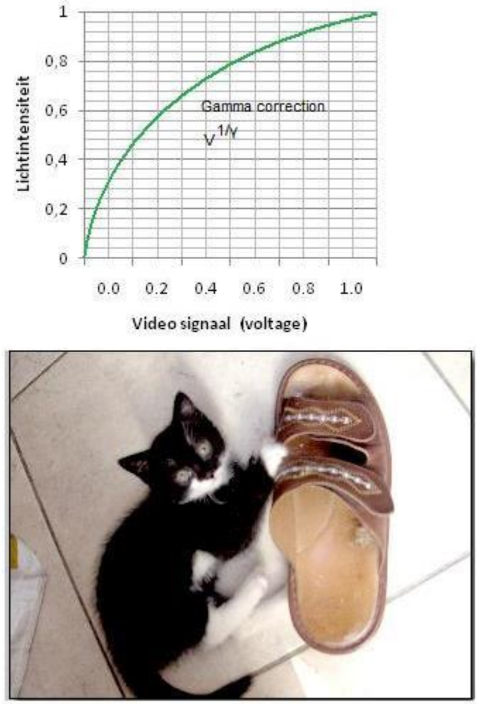
- To correct this non-linear transformation, gamma correction was done:

$$\boxed{V_{out} = \left(\frac{V_{in}}{255}\right)^{\gamma} \cdot 255} \qquad (V_{in}, V_{out} \in \{0,1,\dots,255\})$$

- This is, of course, also applicable for image enhancements.

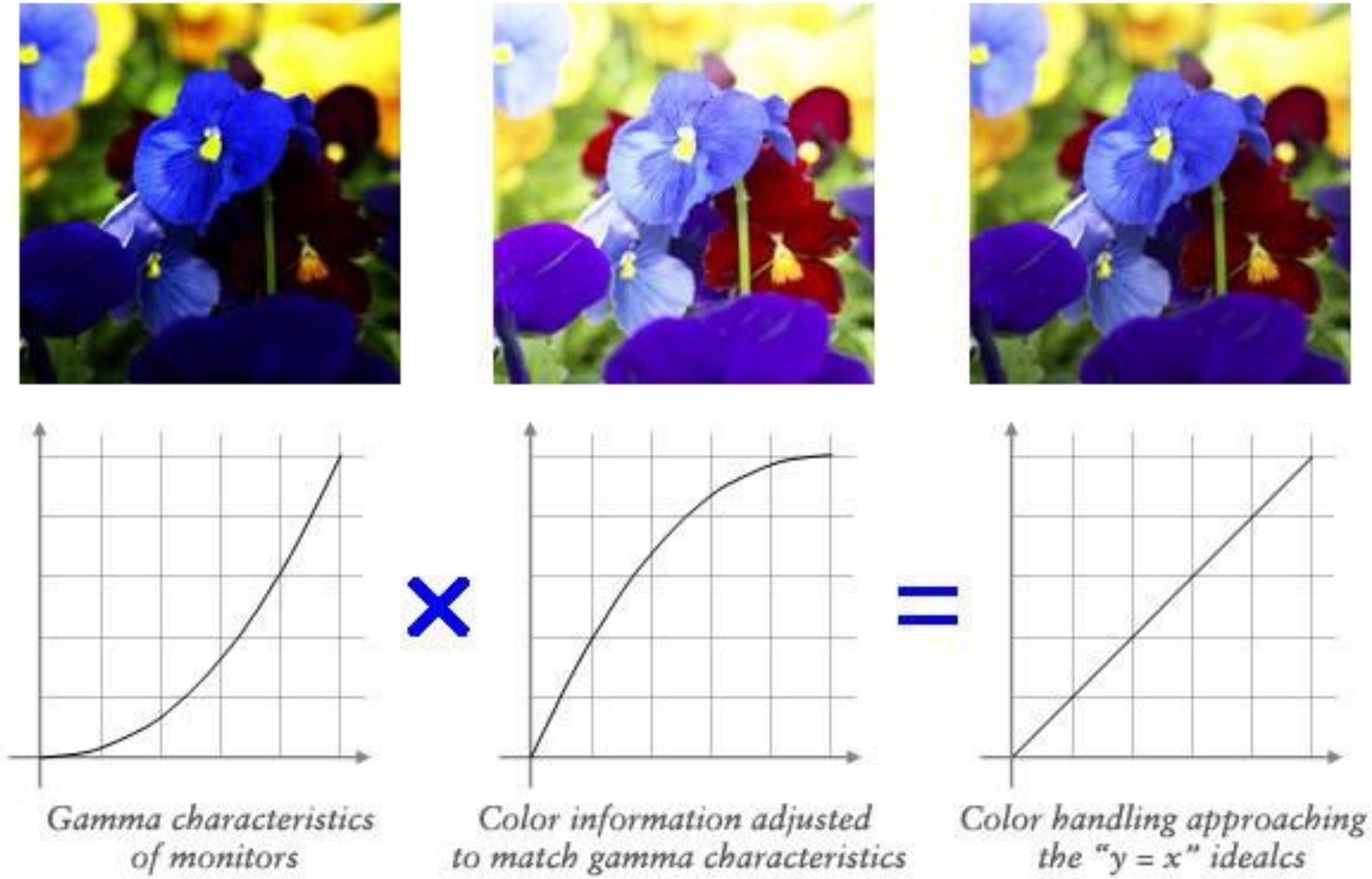


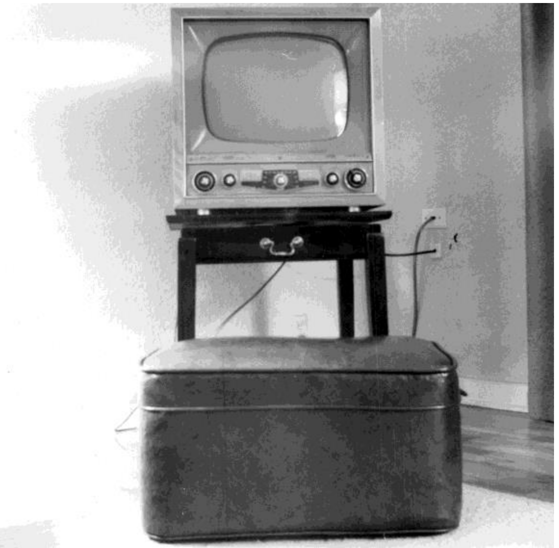




Gamma correction

- Originally, Due to non-linearities in the old CRT televisions, intensities was seen different then they are.





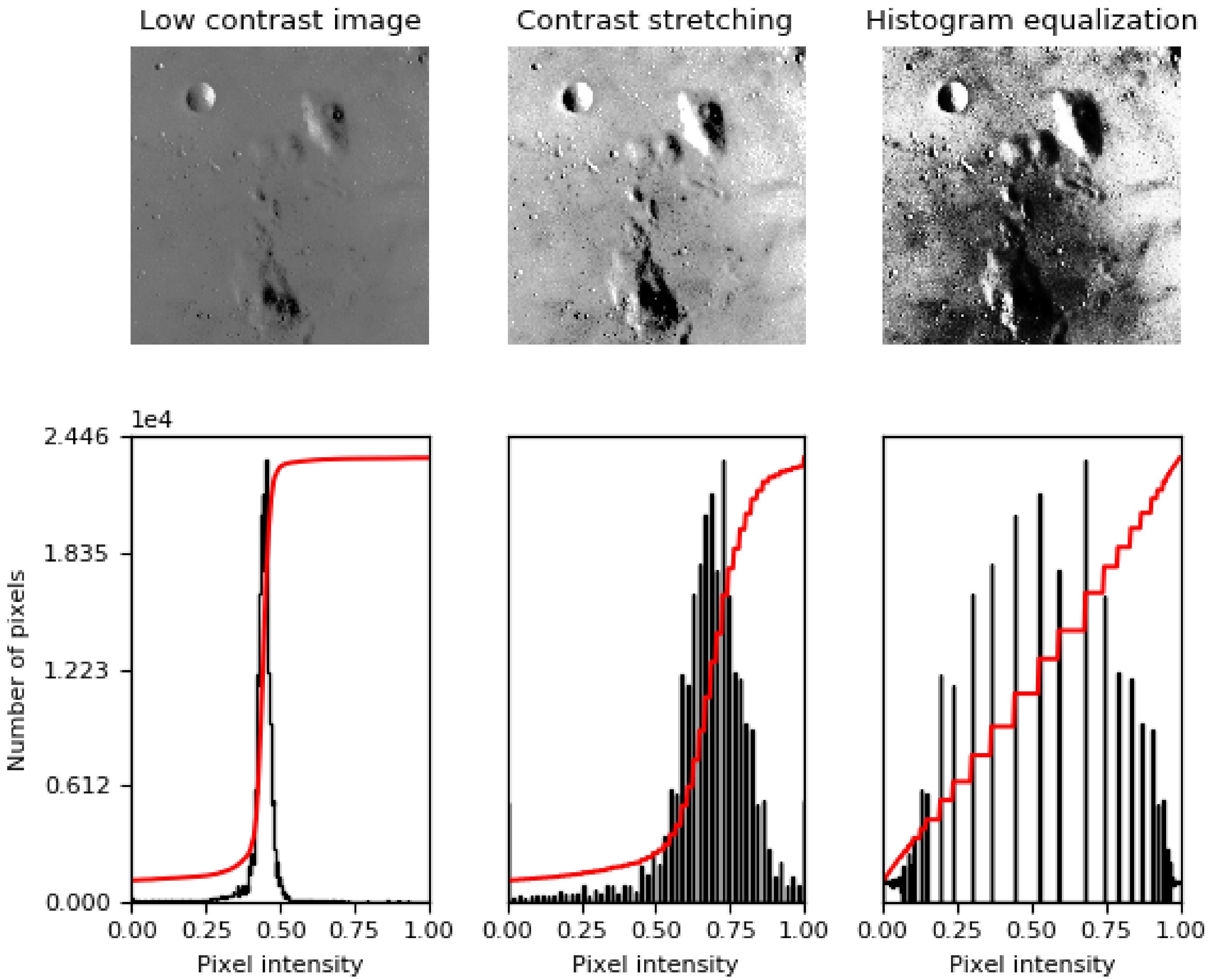
Some more point- wise operators

contents

- Image representation
- Pixel-wise operations
- **Histogram equalization**
- Template matching
- Morphology operators
- Connected components
- Color space

Histogram equalization

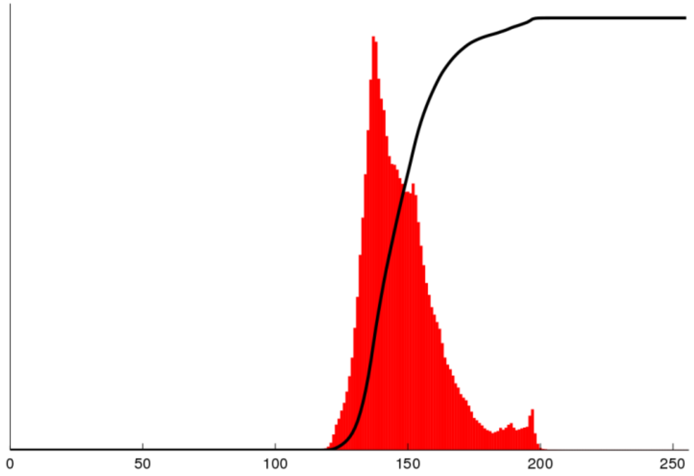
- **Histogram equalization** is a method in image processing of contrast adjustment using the image's histogram.
- This method is used to increase the global contrast of an image and is useful in images with backgrounds and foregrounds that are both bright or both dark.
- **Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.**



Low contrast image
Contrast stretching
Histogram equalization
1e4
2.446
1.835
1.223
0.612
0.000
Number of pixels
0.00
0.25
0.50
0.75
1.00
Pixel intensity

Histogram equalization

- A histogram is a discrete form representation of the distribution of numerical data.
- We will assume at first that our image is continues in the range [0,255] for better understanding.
- Instead of a histogram we will talk about the **probability density function** (**PDF**) $f_X(x)$ of the data.



Reminder: PDF and CDF

- **cumulative distribution function (CDF)** of a real-valued random variable X is the probability that X will take a value less than or equal to x:

$$F_X(x) = P(X \leq x)$$

- Properties of CDF:
  - $\lim_{x \to -\infty} F_X(x) = 0, \quad \lim_{x \to +\infty} F_X(x) = 1$
  - Monotonically non decreasing.
- The **probability density function (PDF)** of a continuous random variable can be determined from the cumulative distribution function by differentiating.

$$f_X(x) = \frac{dF_X(x)}{dx} \quad \text{OR} \quad F_X(x) = \int_{-\infty}^{x} f_X(t)\,dt$$

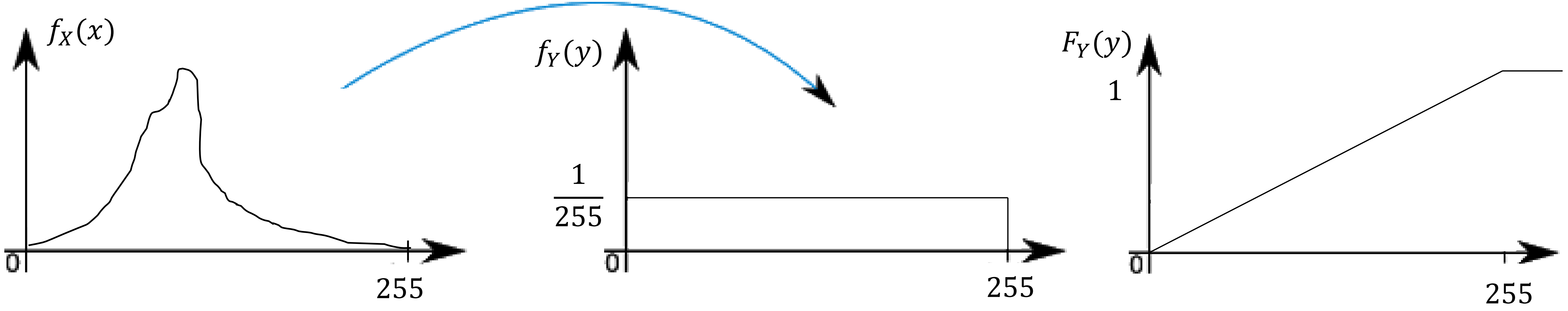
PDF definition- Wikipedia

- **Probability density function (PDF)** can be interpreted as providing a ”relative likelihood” that the value of the random variable would equal that sample.
  - The *absolute likelihood* for a continuous random variable to take on any particular value is 0 (since there are an infinite set of possible values to begin with).
- In a more precise sense, the PDF is used to specify the probability of the random variable falling within a particular range of values. This probability is given by the integral of this variable's PDF over that range and is actually the **CDF.**

PDF equalization

- We want that our resulting PDF [$f_Y(y)$] will be constant for any value in the range [0,255].
- If the PDF is constant, that means that the CDF is linear in [0,255] (integration of constant is a linear function), and so we get the final CDF as:

$$F_Y(y) = P(Y \leq y) = \begin{cases} 0 & : \ y < 0 \\ \frac{y}{255} & : \ 0 \leq y \leq 255 \\ 1 & : \ y > 255 \end{cases}$$



PDF equalization

- So we are looking for a transformation function of the random variable X to Y such that:

$$Y = T(X)$$

- In the interesting area [0,255]:

$$P(Y \leq y) = \frac{y}{255}$$

PDF equalization

- So we are looking for a transformation function of the random variable X to Y such that:

$$Y = T(X)$$

- In the interesting area [0,255]:

$$P(Y \leq y) = \frac{y}{255}$$

$$255 \cdot P(T(X) \leq y) = y$$

PDF equalization

- So we are looking for a transformation function of the random variable X to Y such that:

$$Y = T(X)$$

- In the interesting area [0,255]:

$$P(Y \leq y) = \frac{y}{255}$$

$$255 \cdot P(T(X) \leq y) = y$$

$$255 \cdot P\left(X \leq T^{-1}(y)\right) = y \qquad (assuming\ T\ is\ invertible)$$

PDF equalization

- So we are looking for a transformation function of the random variable X to Y such that:

$$Y = T(X)$$

- In the interesting area [0,255]:

$$P(Y \leq y) = \frac{y}{255}$$

$$255 \cdot P(T(X) \leq y) = y$$

$$255 \cdot P\left(X \leq T^{-1}(y)\right) = y \quad (assuming\ T\ is\ invertible)$$

$$255 \cdot P(X \leq z) = \mathrm{T}(\mathrm{z}) \quad (change\ of\ variables\ z = T^{-1}(y))$$

PDF equalization

- So we are looking for a transformation function of the random variable X to Y such that:

$$Y = T(X)$$

- In the interesting area [0,255]:

$$P(Y \leq y) = \frac{y}{255}$$
$$255 \cdot P(T(X) \leq y) = y$$
$$255 \cdot P\left(X \leq T^{-1}(y)\right) = y \quad (assuming\ T\ is\ invertible)$$
$$255 \cdot P(X \leq z) = \mathrm{T(z)} \quad (change\ of\ variables\ z = T^{-1}(y))$$
$$\boxed{T(x) = F_X(x) \cdot 255}$$

- In fact T is invertible since F_X is Monotonically non decreasing.

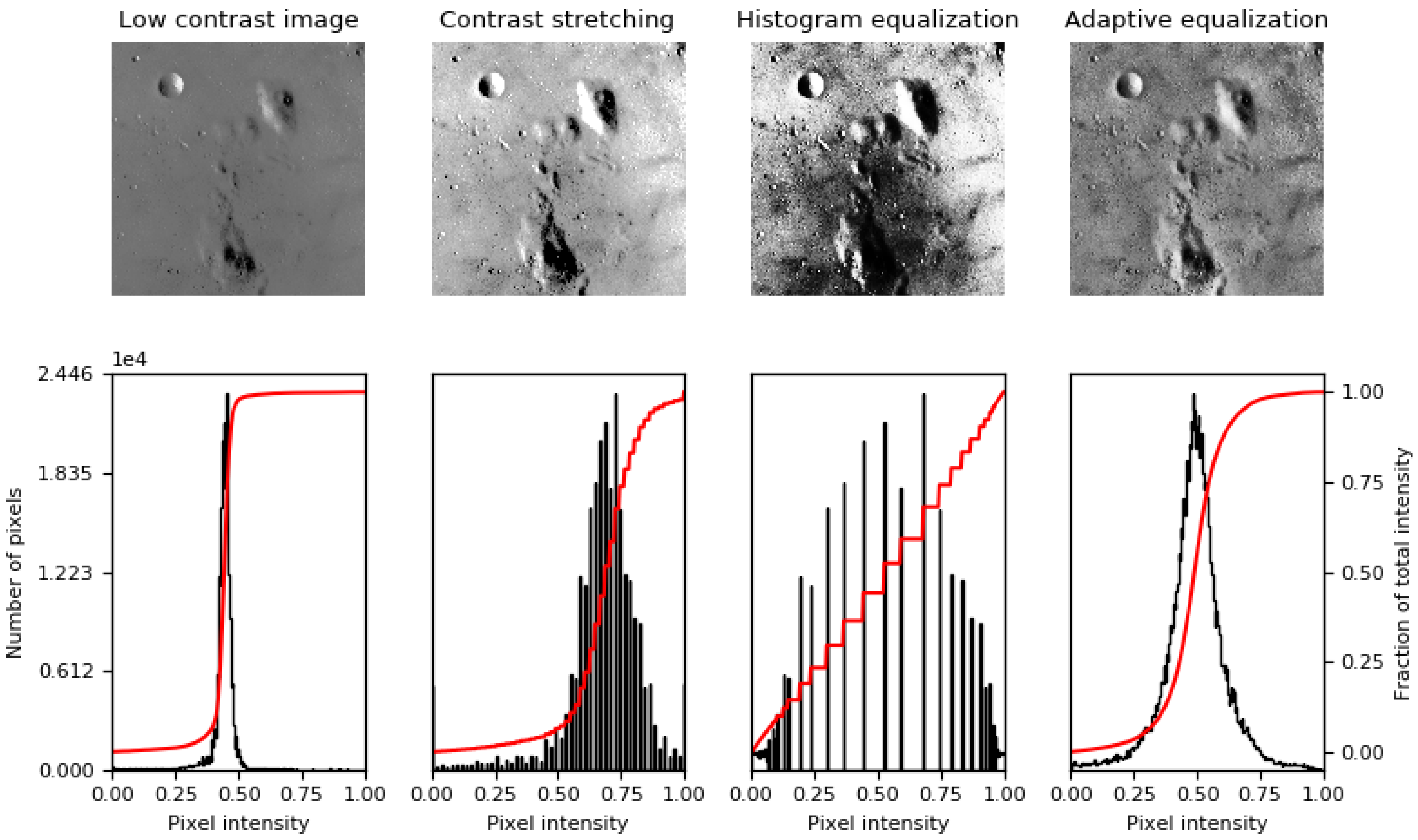
Back to histogram equalization

- The same result is also applicable for discrete space like actual images and their histograms.

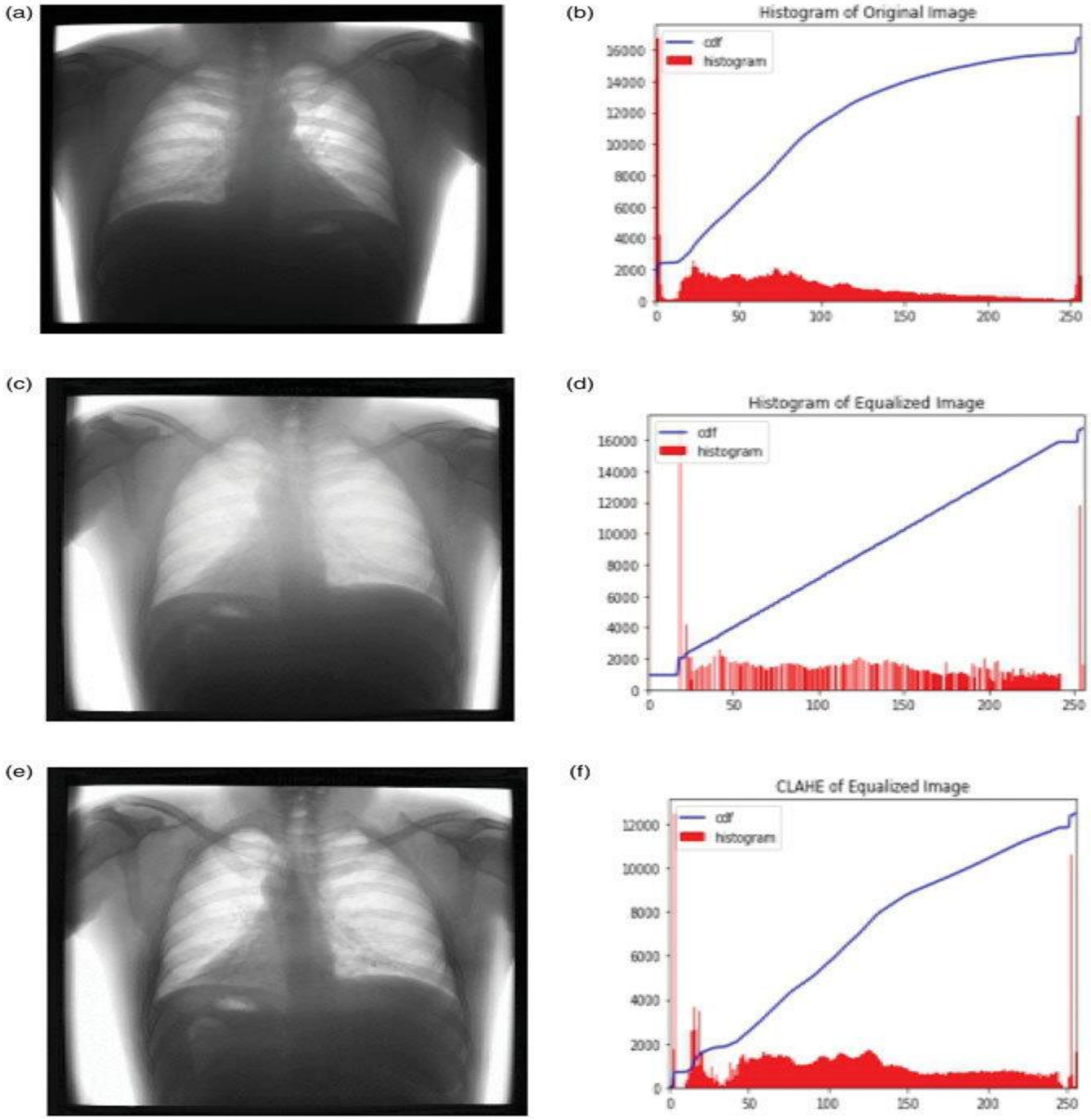
- Build a histogram of a given image.
- To make the histogram act like a discrete PDF- divide each bin by the sum of all bins.
- Cumulative sum the PDF to get the discrete CDF.
- Un-normalize the CDF and round the results back to uint8:

$$f_{eq}(x) = round(CDF(x) \cdot 255)$$

Other variants of histogram equalization



Other variants of histogram equalization

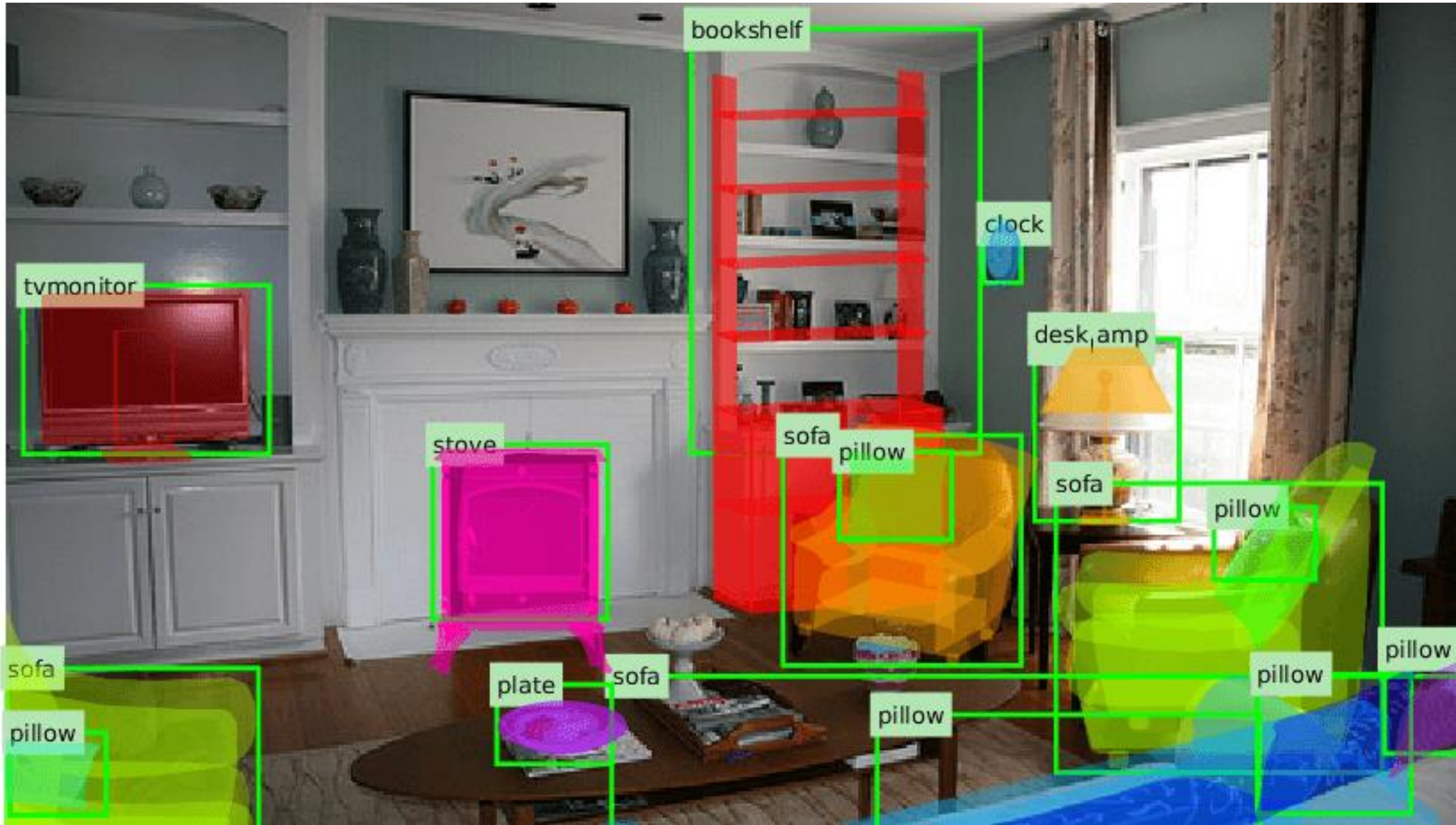


contents

- Image representation
- Pixel-wise operations
- Histogram equalization
- **Template matching**
- Morphology operators
- Connected components
- Color space

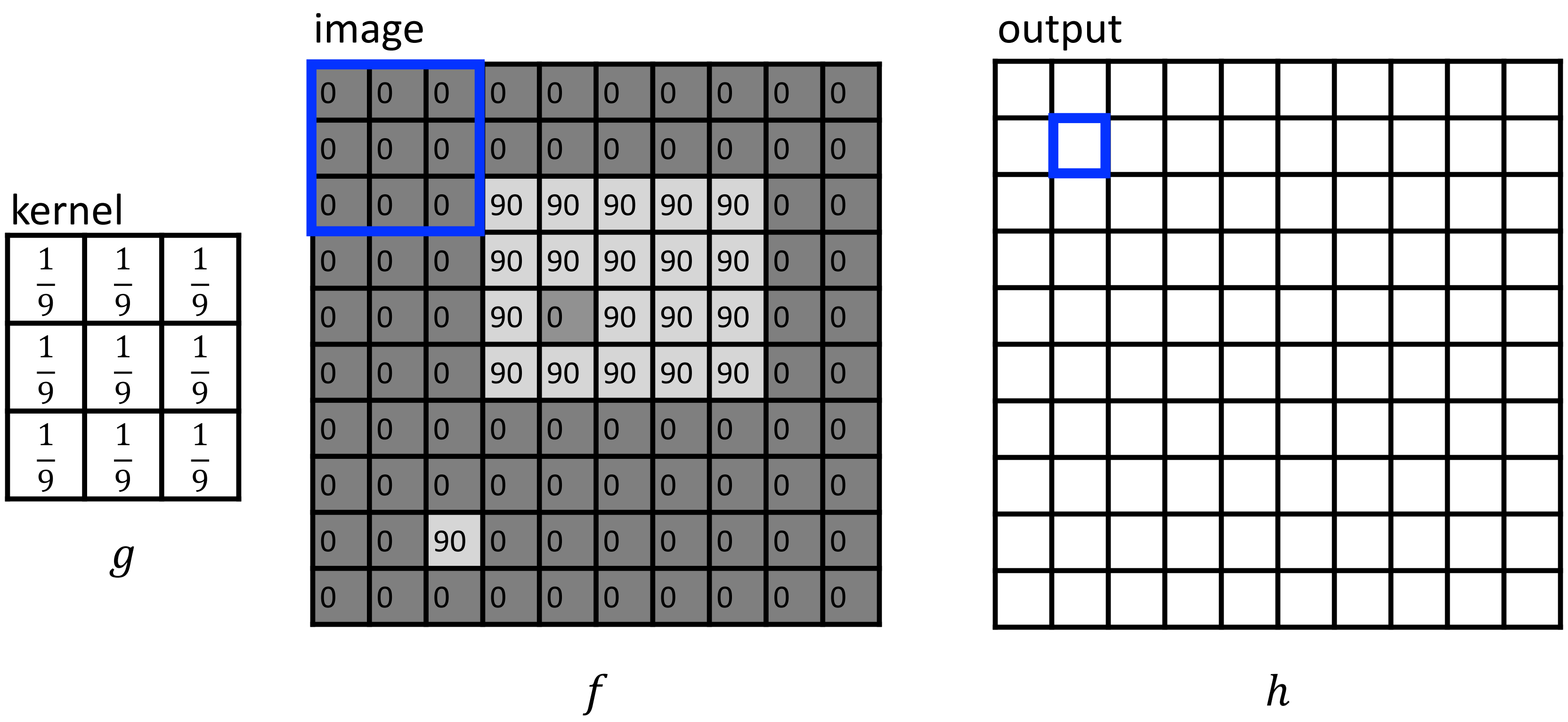
Template matching

- Given an image template- find it in another image.
- Template matching is a sub-field in **object recognition**.
  - We will see it **a lot** of this topic in this course:
    - Cross correlation
    - Feature based – SIFT
    - Neural networks



Example output

First- let’s understand what is cross-correlation



Run the filter

kernel

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

g

image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

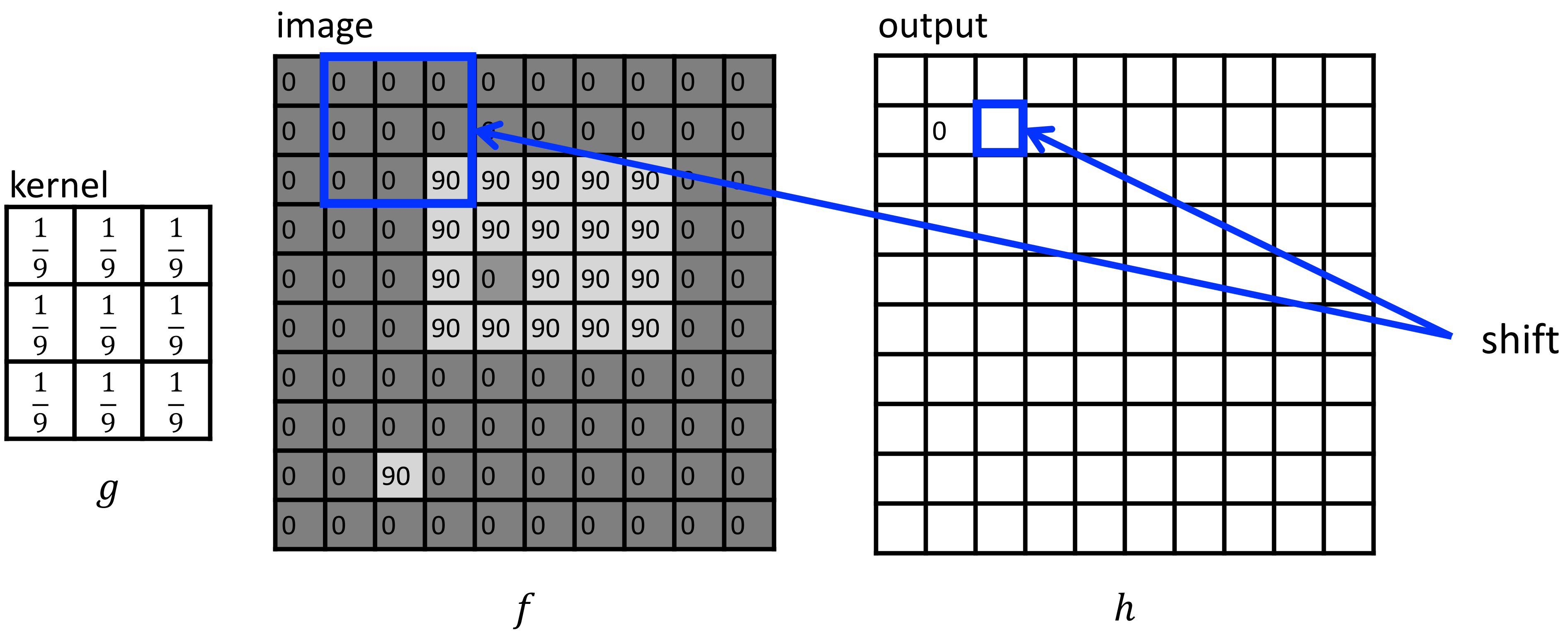
f

output

	0								

h

Run the filter



Run the filter

kernel

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

g

image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

f

output

	0	10							

h

Run the filter

kernel

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

g

image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

f

output

	0	10	20						

h

Run the filter

kernel

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

g

image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

f

output

	0	10	20	30					

h

Run the filter

kernel

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

g

image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

f

output

	0	10	20	30	$\boxed{30}$				

h

Run the filter

kernel

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

g

image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

f

output

	0	10	20	30	30	30			

h

… and the result is

kernel

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

g

image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

f

output

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	0	10	20	30	30	30	20	10	
	10	10	10	10	0	0	0	0	
	10	10	10	10	0	0	0	0	

h

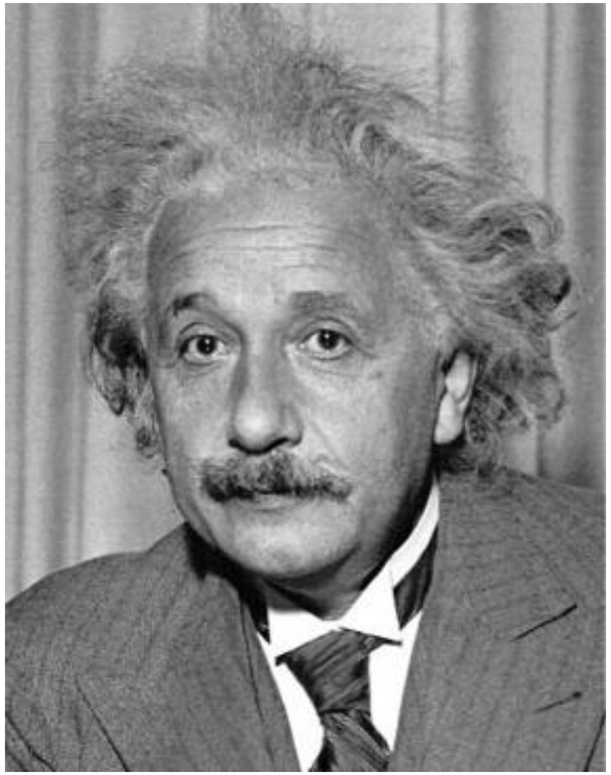
Cross correlation can also be more simply denoted as $h = g \star f$

The full mathematical notation is this:

$$h[m,n] = \sum_{k=-1}^{1} \sum_{l=-1}^{1} g[k,l] f[m+k, n+l]$$

CC – cross correlation

Can cross-correlation be good for template matching? Let’s take our template  that we want to find as our kernel



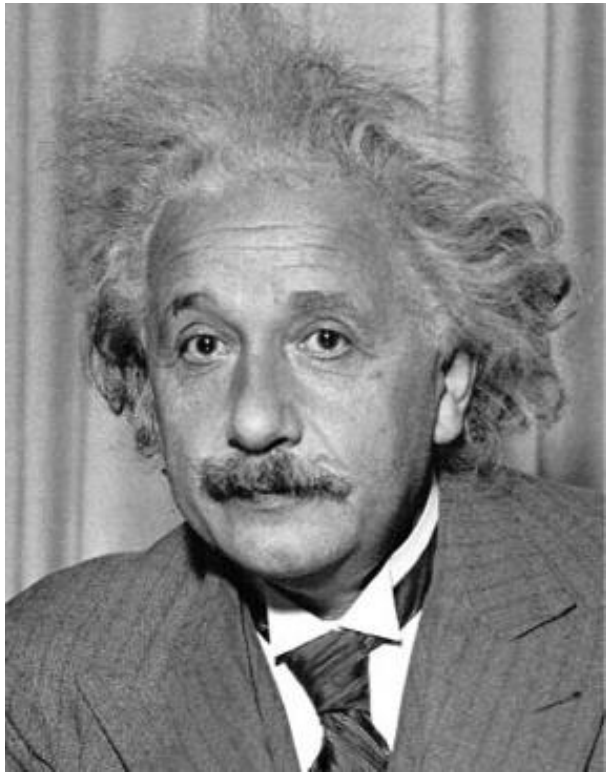
output

$$h[m,n] = \sum_{k=-1}^{1} \sum_{l=-1}^{1} g[k,l] f[m+k, n+l]$$

image

What will the output look like?

CC – cross correlation



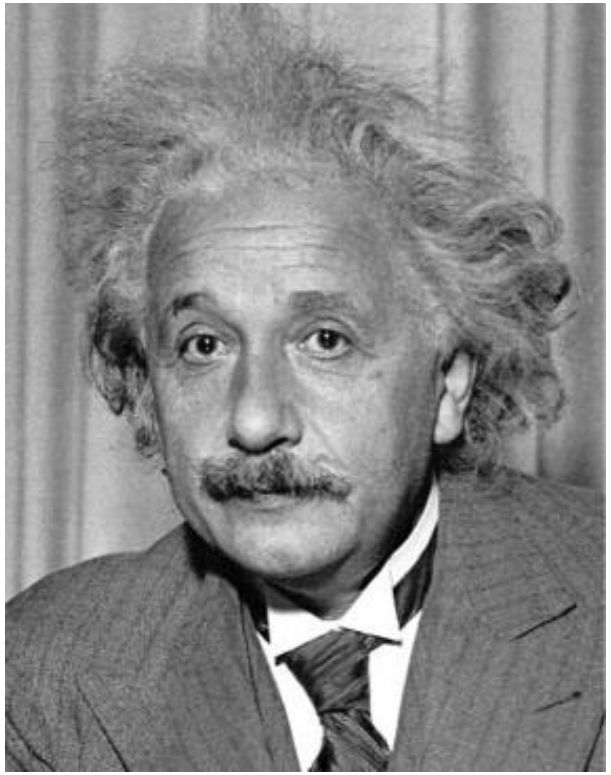
output

$$h[m,n] = \sum_{k=-1}^{1} \sum_{l=-1}^{1} g[k,l] f[m+k, n+l]$$

image

What will the output look like?

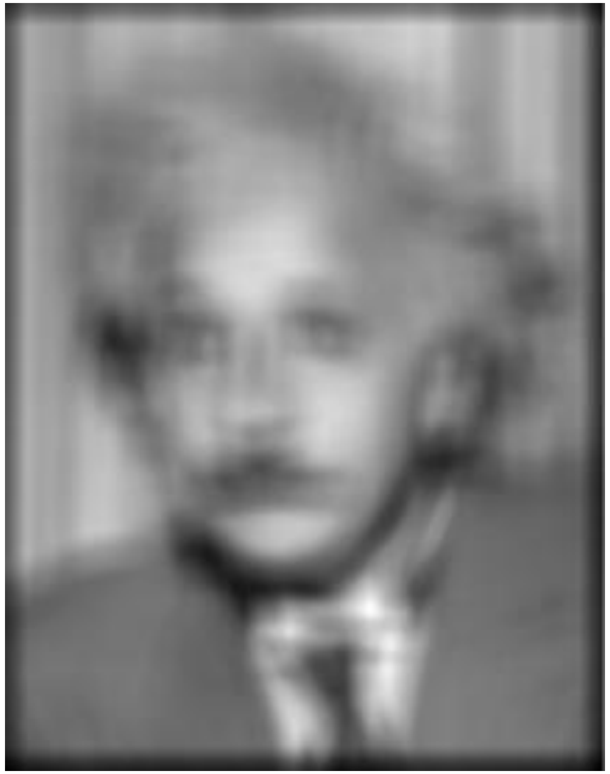
CC – cross correlation



output

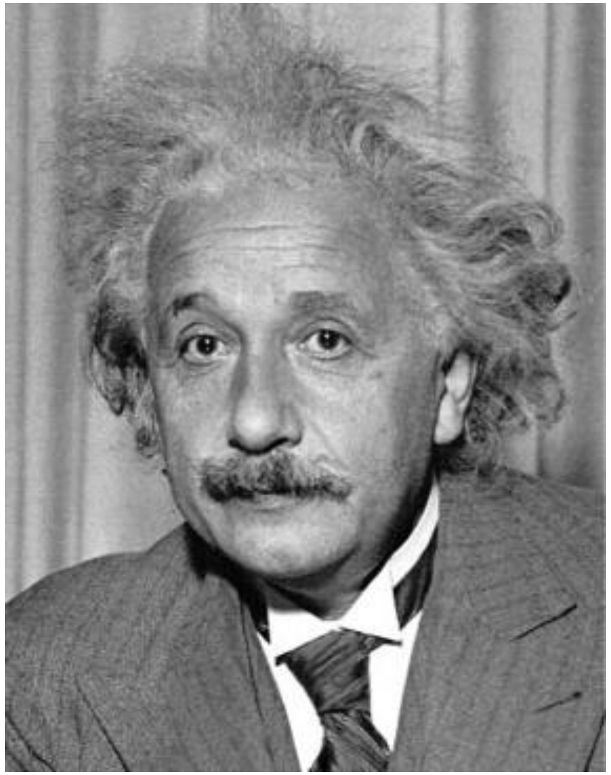
$$h[m,n] = \sum_{k=-1}^{1} \sum_{l=-1}^{1} g[k,l] f[m+k, n+l]$$

image



Is this good for
template matching?

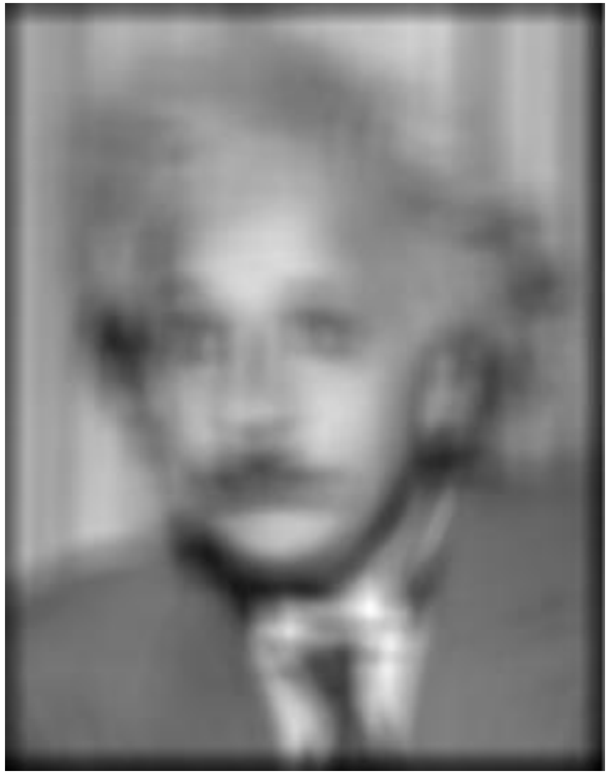
CC – cross correlation



output

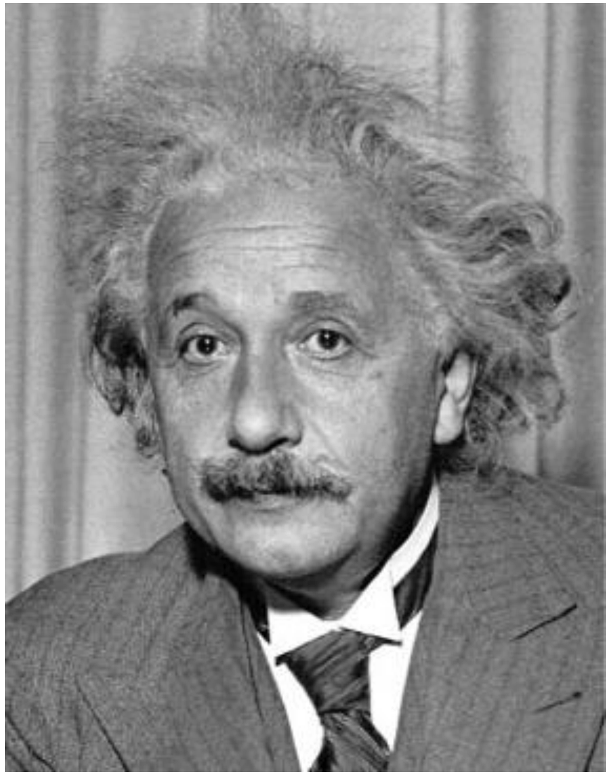
$$h[m,n] = \sum_{k=-1}^{1} \sum_{l=-1}^{1} g[k,l] f[m+k, n+l]$$

image



Increases for higher local intensities.

Zero mean cross correlation

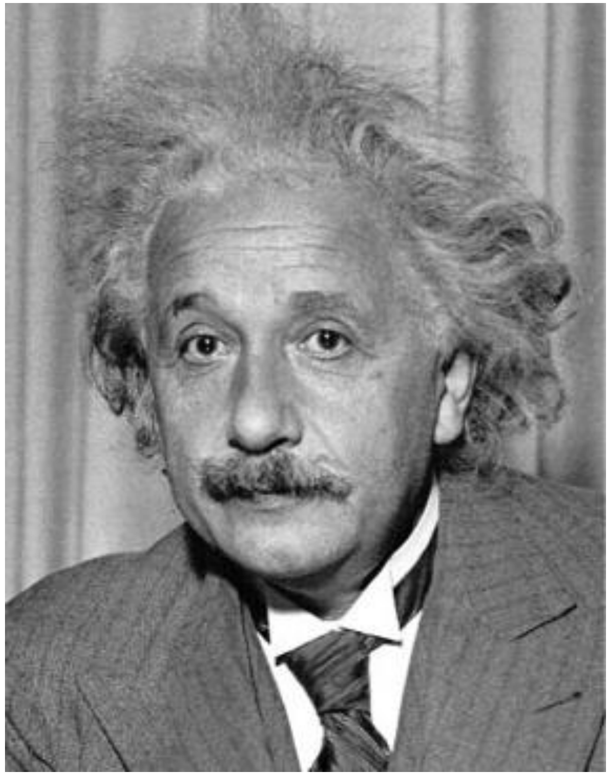


template mean

$$h[m,n] = \sum_{k,l} (g[k,l] - \bar{g}) f[m+k, n+l]$$

What will the output look like?

Zero mean cross correlation

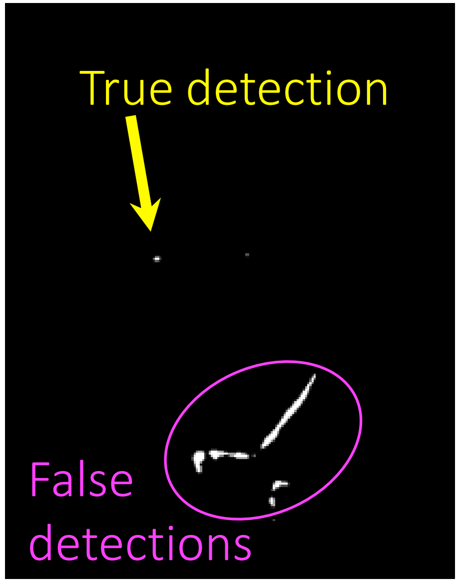


template mean

$$h[m,n] = \sum_{k,l} (g[k,l] - \bar{g}) f[m+k, n+l]$$

output

thresholding



Zero mean CC is good enough for most problems but can also cause false detections in high contrast areas.

CC

$$\underbrace{\begin{array}{|c|c|c|}\hline 255 & 255 & 255 \\ \hline 255 & 0 & 255 \\ \hline 255 & 255 & 255 \\ \hline \end{array}}_{g} \;\star\; \underbrace{\begin{array}{|c|c|c|}\hline 255 & 255 & 255 \\ \hline 255 & 255 & 255 \\ \hline 255 & 255 & 255 \\ \hline \end{array}}_{f} \;\longrightarrow$$

$$\begin{array}{|c|c|c|}\hline 255^2 & 255^2 & 255^2 \\ \hline 255^2 & 0 & 255^2 \\ \hline 255^2 & 255^2 & 255^2 \\ \hline \end{array} \;\longrightarrow\; \underbrace{\boxed{8 * 255^2}}_{h}$$ 😥

1. Scalar multiplication

2. Summation

Zero mean CC

28	28	28
28	-227	28
28	28	28

$g - \bar{g}$

★

255	255	255
255	255	255
255	255	255

f

→

7140	7140	7140
7140	−57855	7140
7140	7140	7140

→

-735

h

😮

1. Scalar multiplication

2. Summation

Zero mean CC

28	28	28
28	-227	28
28	28	28

$g - \bar{g}$

★

255	255	255
255	0	255
255	255	255

f

→

7140	7140	7140
7140	0	7140
7140	7140	7140

→

57120

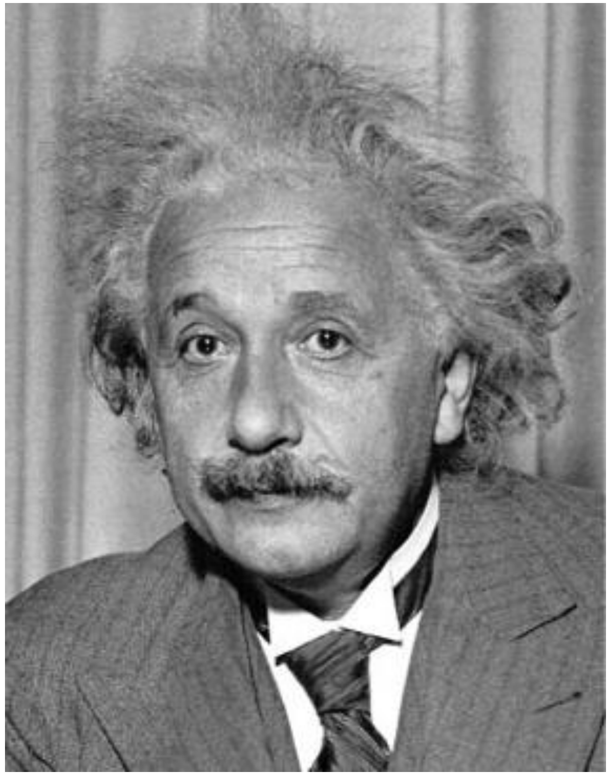
h

😍

1. Scalar multiplication

2. Summation

ZNCC – zero mean normalized cross correlation

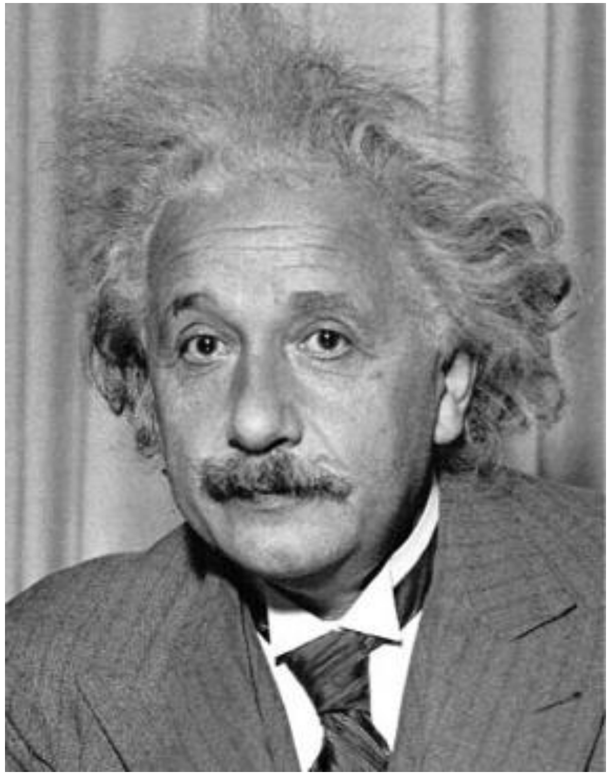


What will the output look like?

$$h[m,n] = \frac{\sum_{k,l}(g[k,l]-\bar{g})(f[m+k,n+l]-\bar{f}_{m,n})}{\sqrt{\left(\sum_{k,l}(g[k,l]-\bar{g})^2 \sum_{k,l}(f[m+k,n+l]-\bar{f}_{m,n})^2\right)}}$$

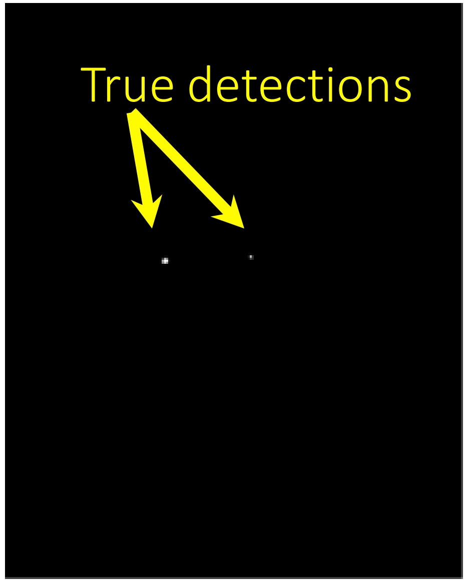
The below square root is the product of both template and patch STD.

ZNCC – zero mean normalized cross correlation

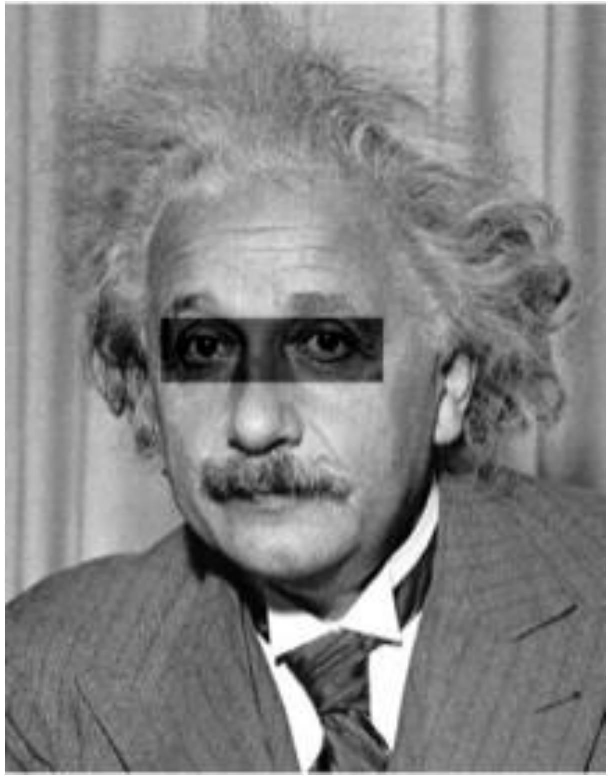


output

thresholding

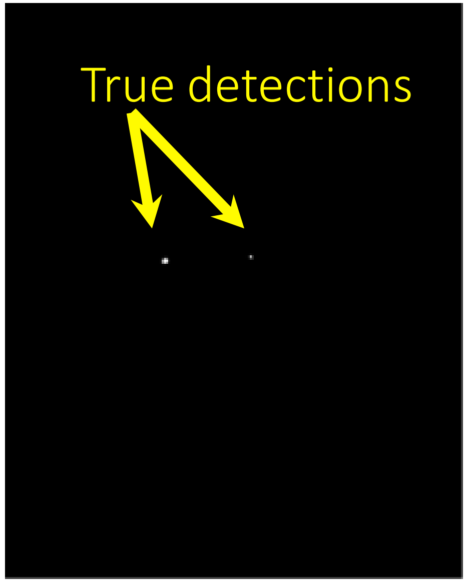


ZNCC – zero mean normalized cross correlation



output

thresholding



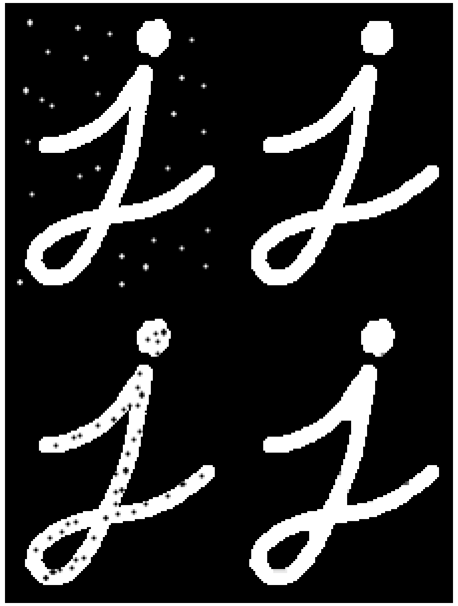
robust to change in intensities

contents

Morphology

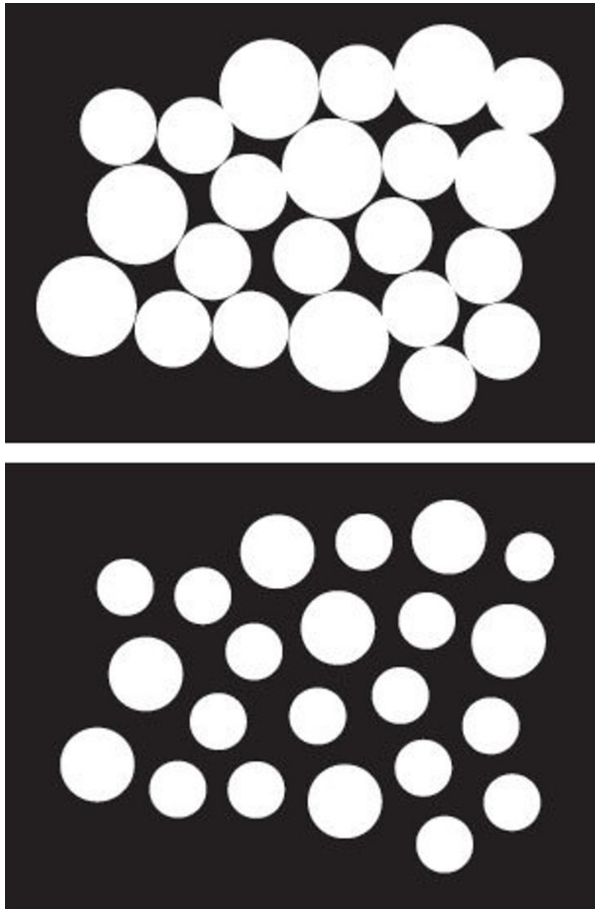
Examples:

Image cleaning

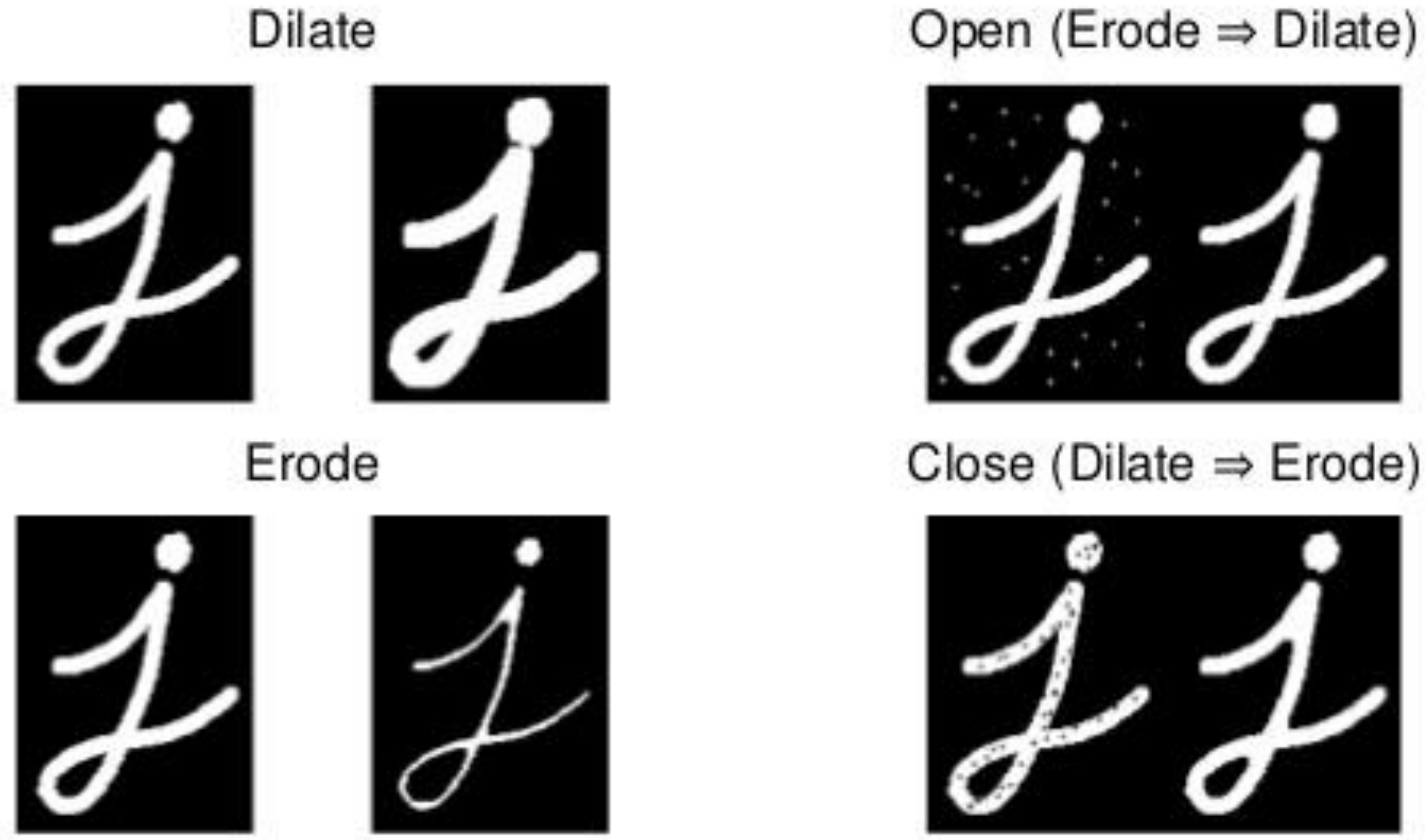


Style

Coin counting
(using connected components)

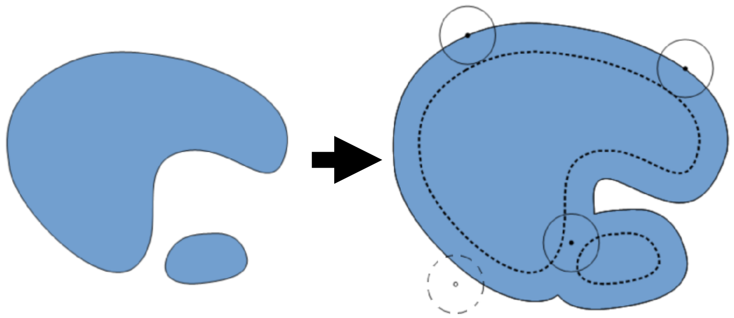


The 4 basic operators

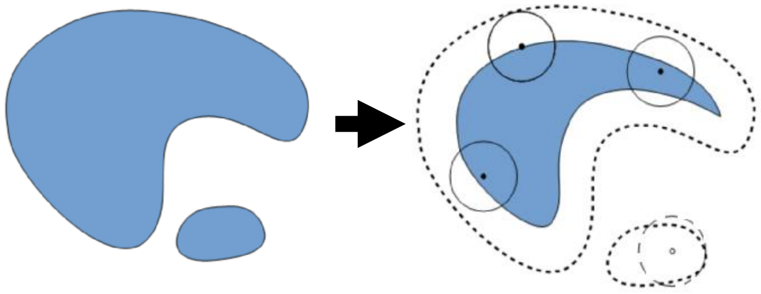


Morphology: geometric interpretation

- Each kernel (g) has an anchor point (usually in the kernel center).
- Dilation: the final shape is all points where the anchor point can be placed in which **the kernel touches a part of the original shape**.



- Erosion: the final shape is all points where the anchor point can be placed in which **all kernel points touch the original shape**.



1. Cross-correlation with the kernel

$$
\underset{f}{\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ \hline
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline
\end{array}}
\star
\underset{g}{\begin{array}{|c|c|c|}
\hline
1 & 1 & 1 \\ \hline
1 & 1 & 1 \\ \hline
1 & 1 & 1 \\ \hline
\end{array}}
=
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
 & & & & & & & \\ \hline
 & 2 & 3 & 3 & 2 & 2 & 1 & \\ \hline
 & 4 & 6 & 6 & 5 & 5 & 3 & \\ \hline
 & 6 & 9 & 9 & 7 & 5 & 2 & \\ \hline
 & 6 & 9 & 9 & 7 & 5 & 2 & \\ \hline
 & 6 & 8 & 7 & 4 & 2 & 0 & \\ \hline
 & 6 & 7 & 5 & 2 & 1 & 0 & \\ \hline
 & & & & & & & \\ \hline
\end{array}
$$

2. Threshold the result

For **dilation-** threshold with 1

	2	3	3	2	2	1	
	4	6	6	5	5	3	
	6	9	9	7	5	2	
	6	9	9	7	5	2	
	6	8	7	4	2	0	
	6	7	5	2	1	0	

$$\geq 1 =$$

	1	1	1	1	1	1	
	1	1	1	1	1	1	
	1	1	1	1	1	1	
	1	1	1	1	1	1	
	1	1	1	1	1	0	
	1	1	1	1	1	0	

2. Threshold the result

For **erosion-** threshold with the sum of the kernel

2	3	3	2	2	1
4	6	6	5	5	3
6	9	9	7	5	2
6	9	9	7	5	2
6	8	7	4	2	0
6	7	5	2	1	0

$$\geq sum(g) =$$

0	0	0	0	0	0
0	0	0	0	0	0
0	1	1	0	0	0
0	1	1	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

$sum(g) = 9$ in this example

Morphology: algorithm

- Each morphology operator is constructed as such:
  1. Select a structure element (binary kernel)
  2. Cross-correlate with input binary image $h = f \star g$
  3. Threshold the output

$$g = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

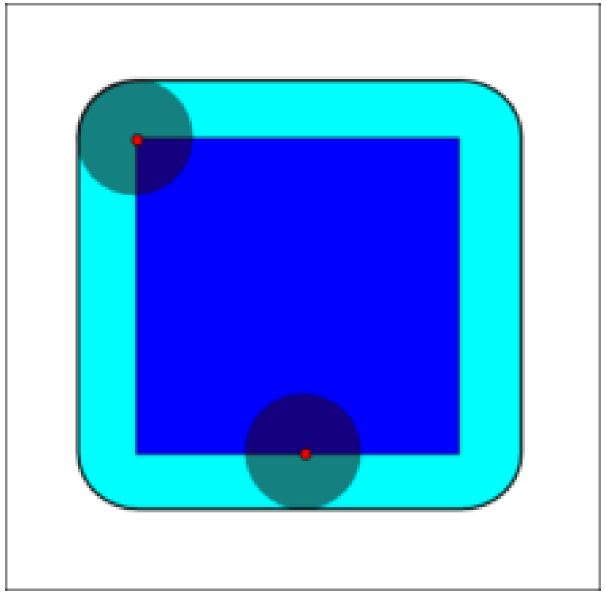
$$\theta_{TH}(x,t) = \begin{cases} 1 & if \quad x \geq t, \\ 0 & else \end{cases}$$

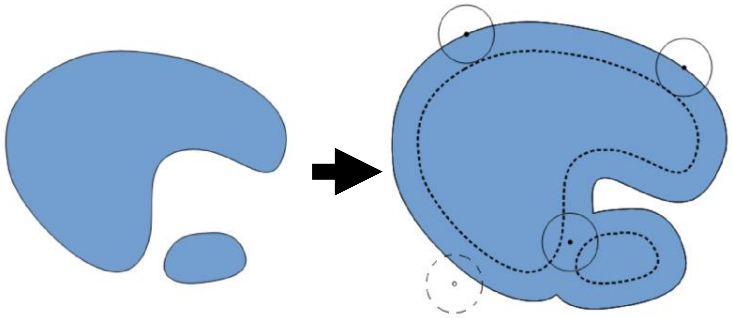
- Overall morphologic operation should look like so:

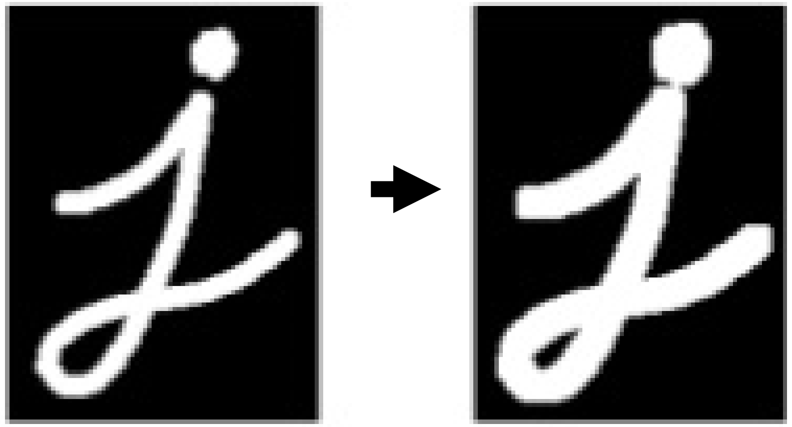
$$k = \theta_{TH}(f \star g, t)$$

Dilation- examples

- $k = \theta_{TH}(f \star g, \boldsymbol{t} = \mathbf{1})$

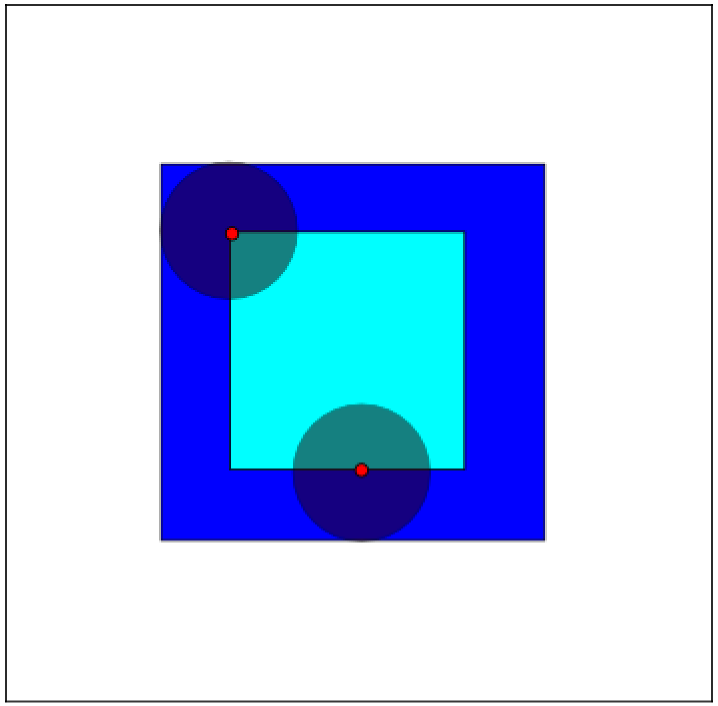


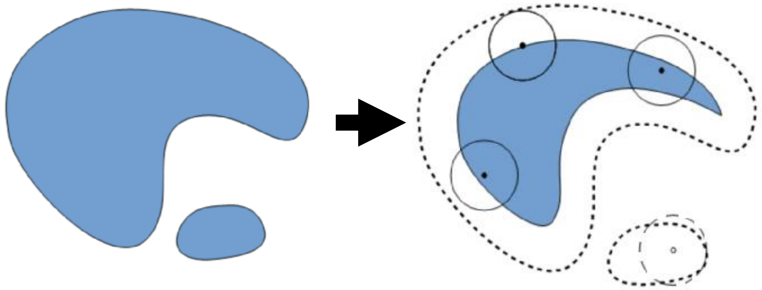


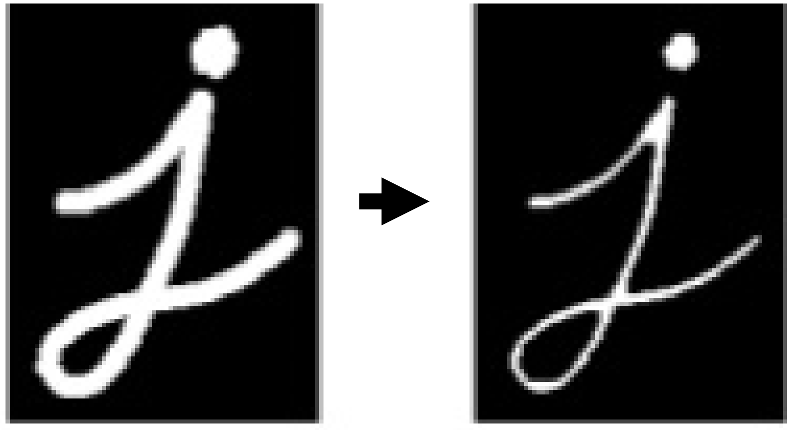


Erosion- examples

- $k = \theta_{TH}(f \star g, \boldsymbol{t} = \boldsymbol{sum}(\boldsymbol{g}))$

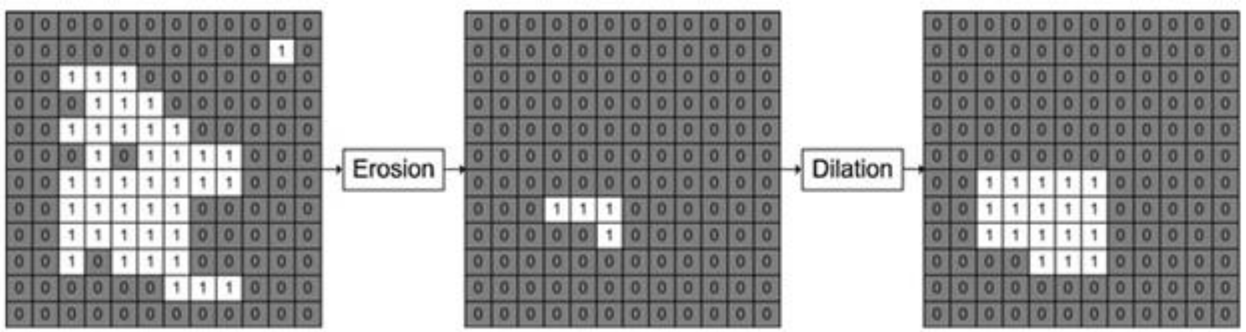




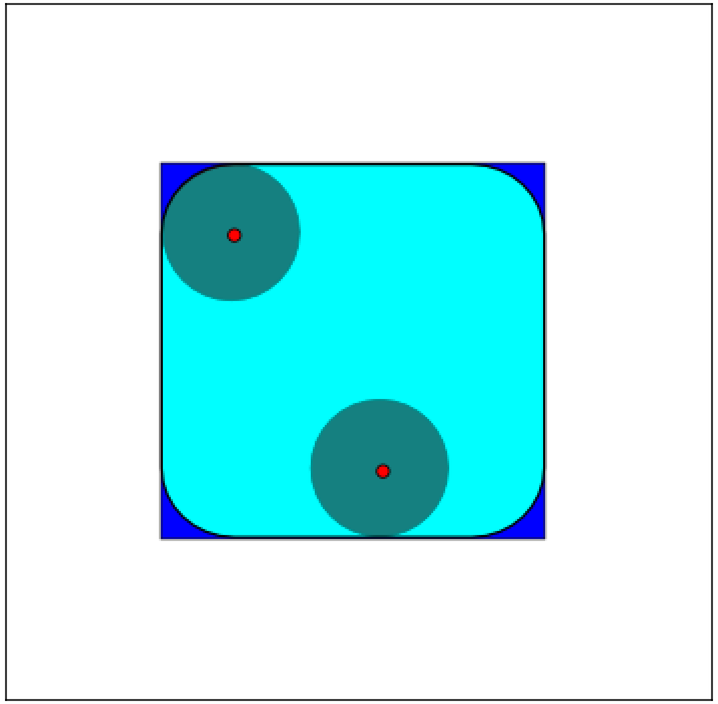


Opening

- Erosion followed by dilation.
  – The effect is of removing noise or sharp edges.

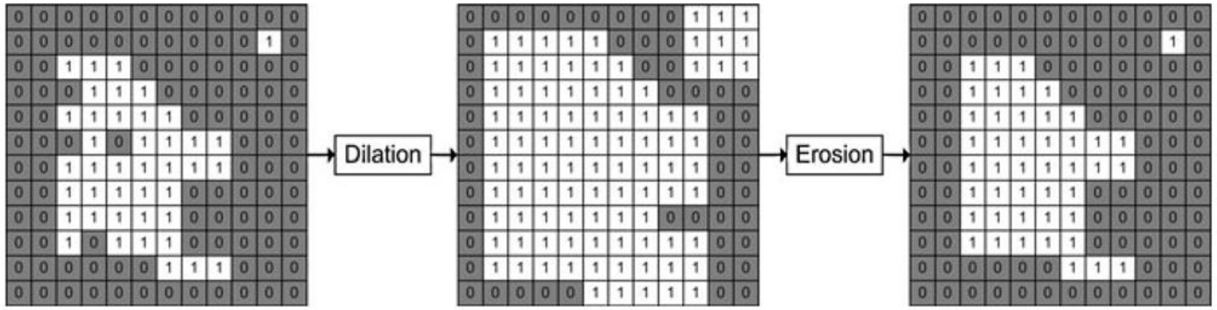


Open (Erode ⇒ Dilate)

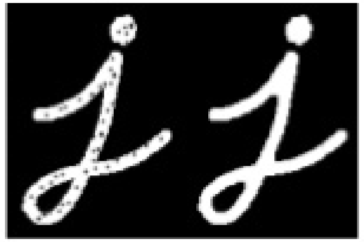


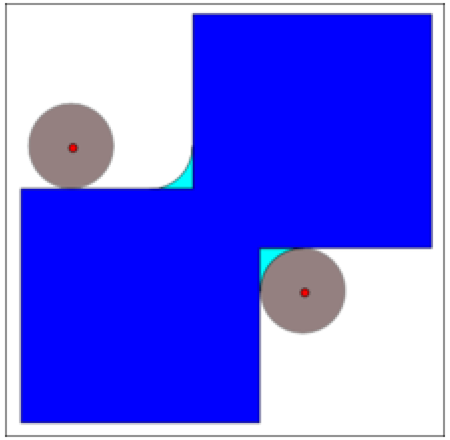
Closing

- Dilation followed by erosion.
  - The effect is of closing of narrow gaps and holes.

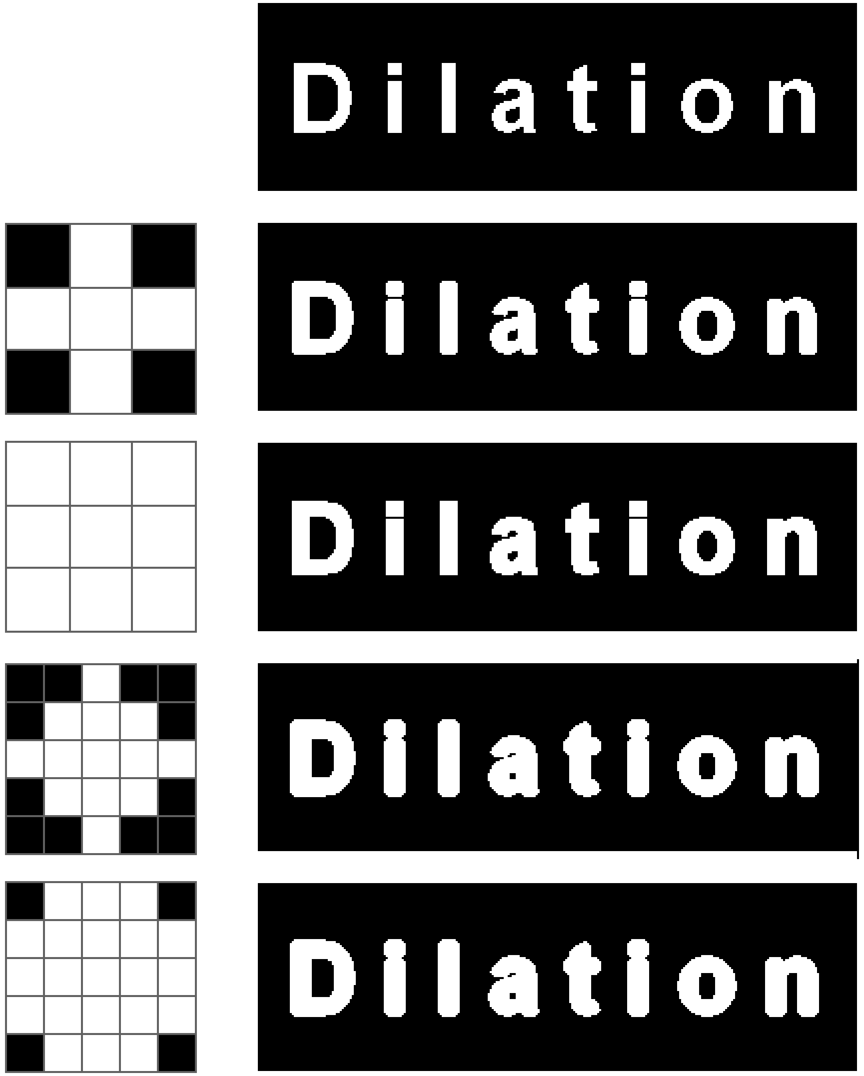


Close (Dilate ⇒ Erode)





Affect of different kernels

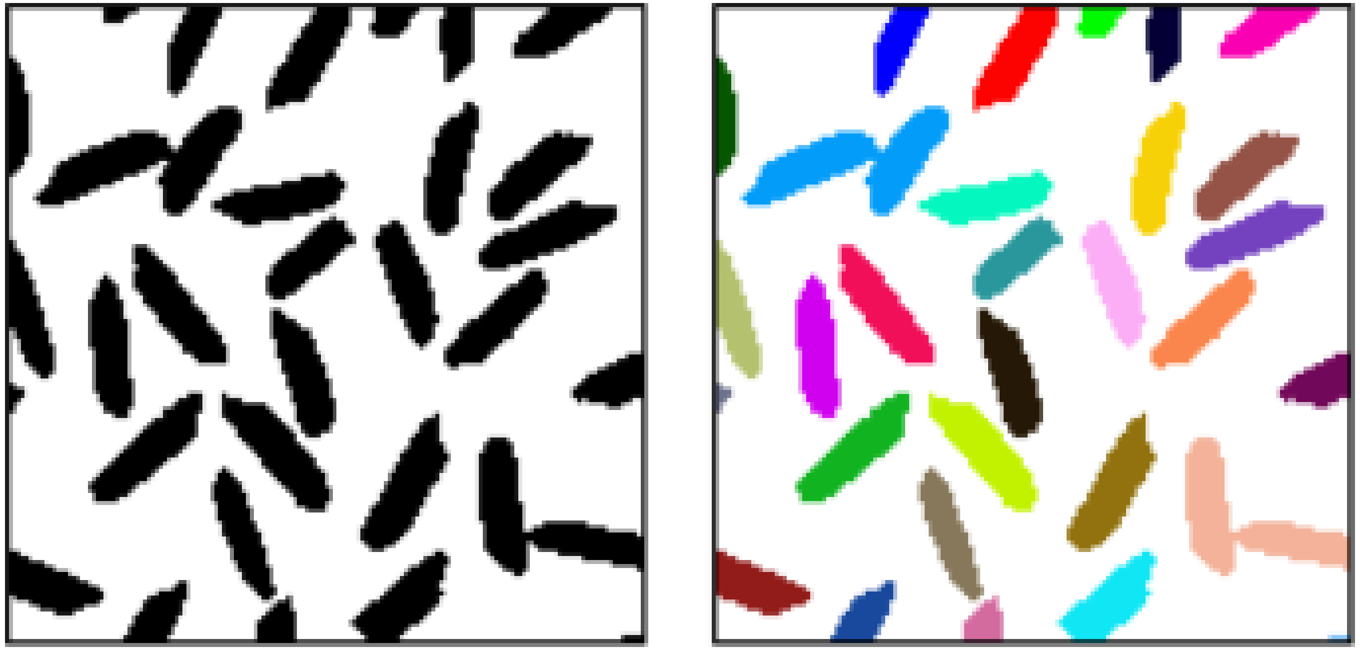


We will see non-symmetrical kernels in the HW

contents

Connected components

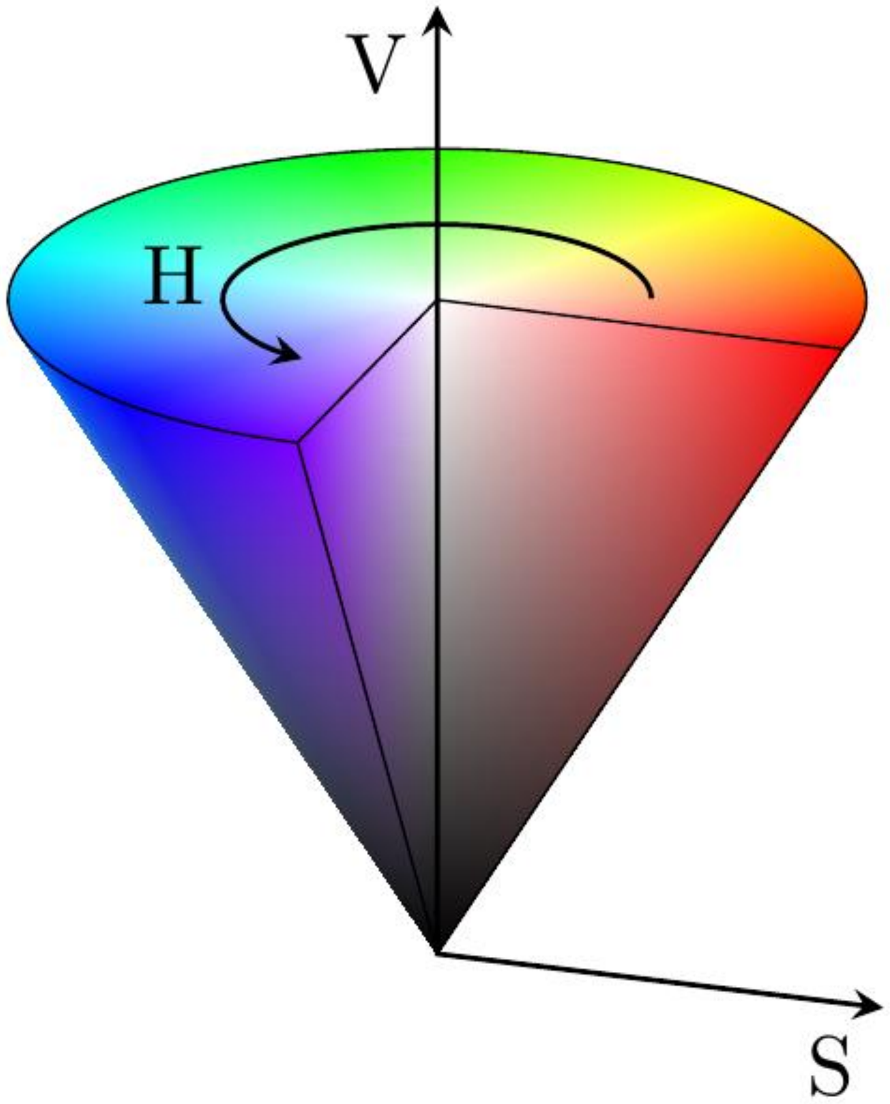
- Defined as regions of adjacent pixels that have the same value.
- Commonly used with binary images to find stand alone objects.
  - e.g.: letters in a document.



contents

- Image representation
- Pixel-wise operations
- Histogram equalization
- Template matching
- Morphology operators
- Connected components
- **Color space**

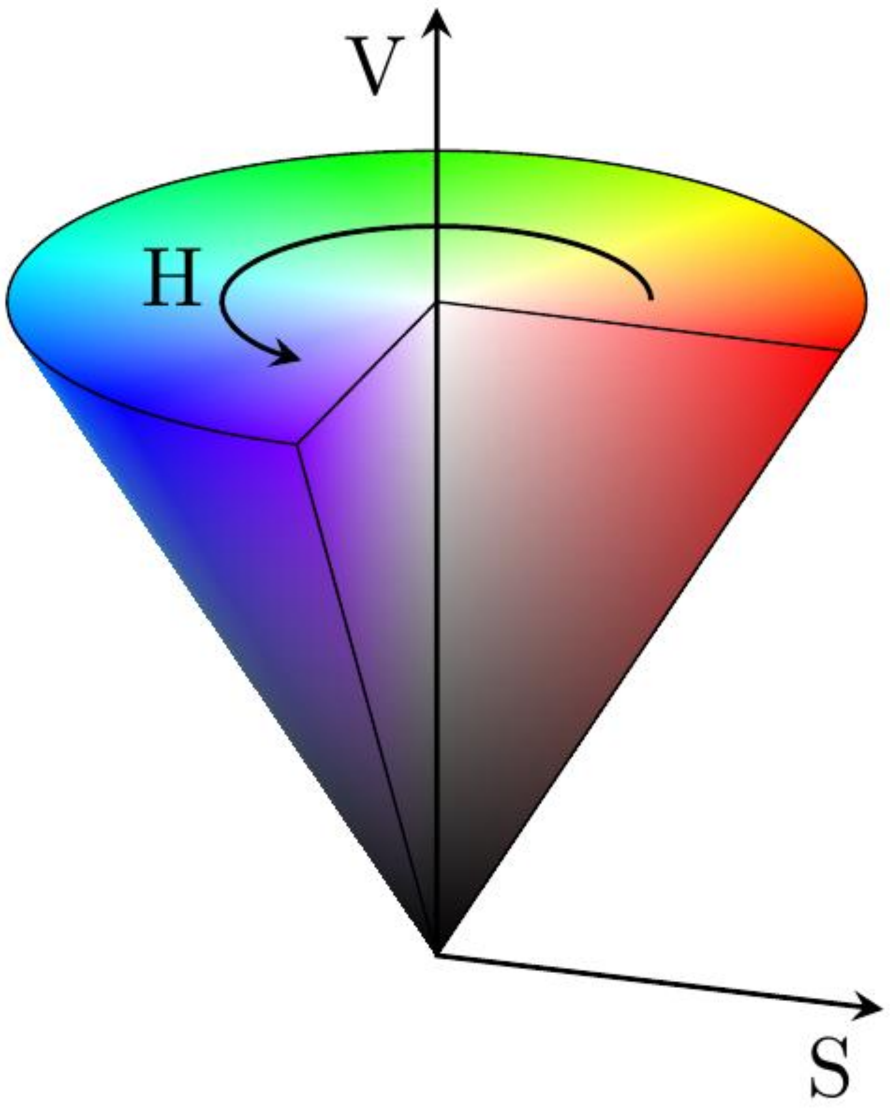
HSV

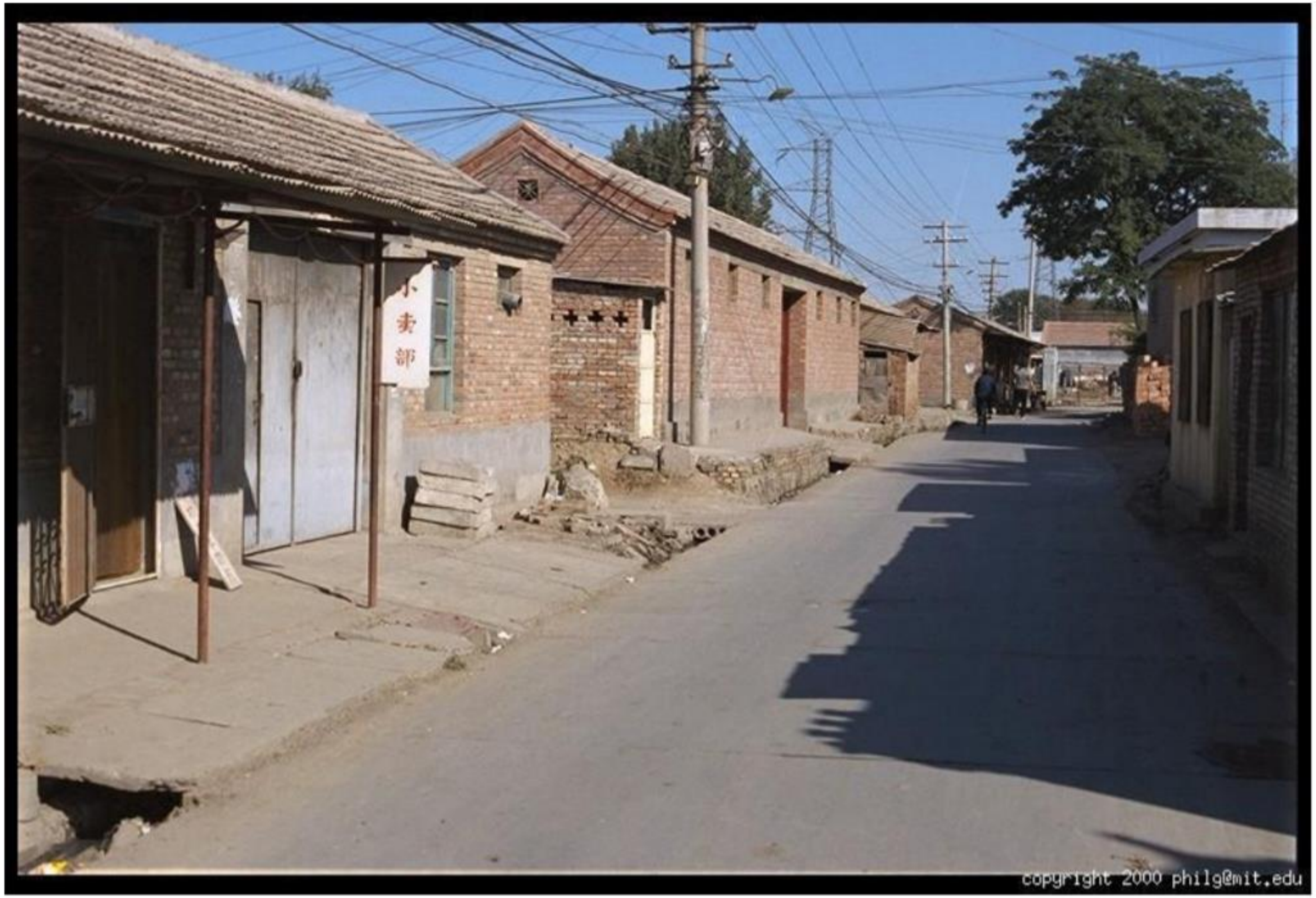


https://cs.brown.edu/courses/csci1430/lectures/2020Spring_08_CamerasLightAndColor.pdf

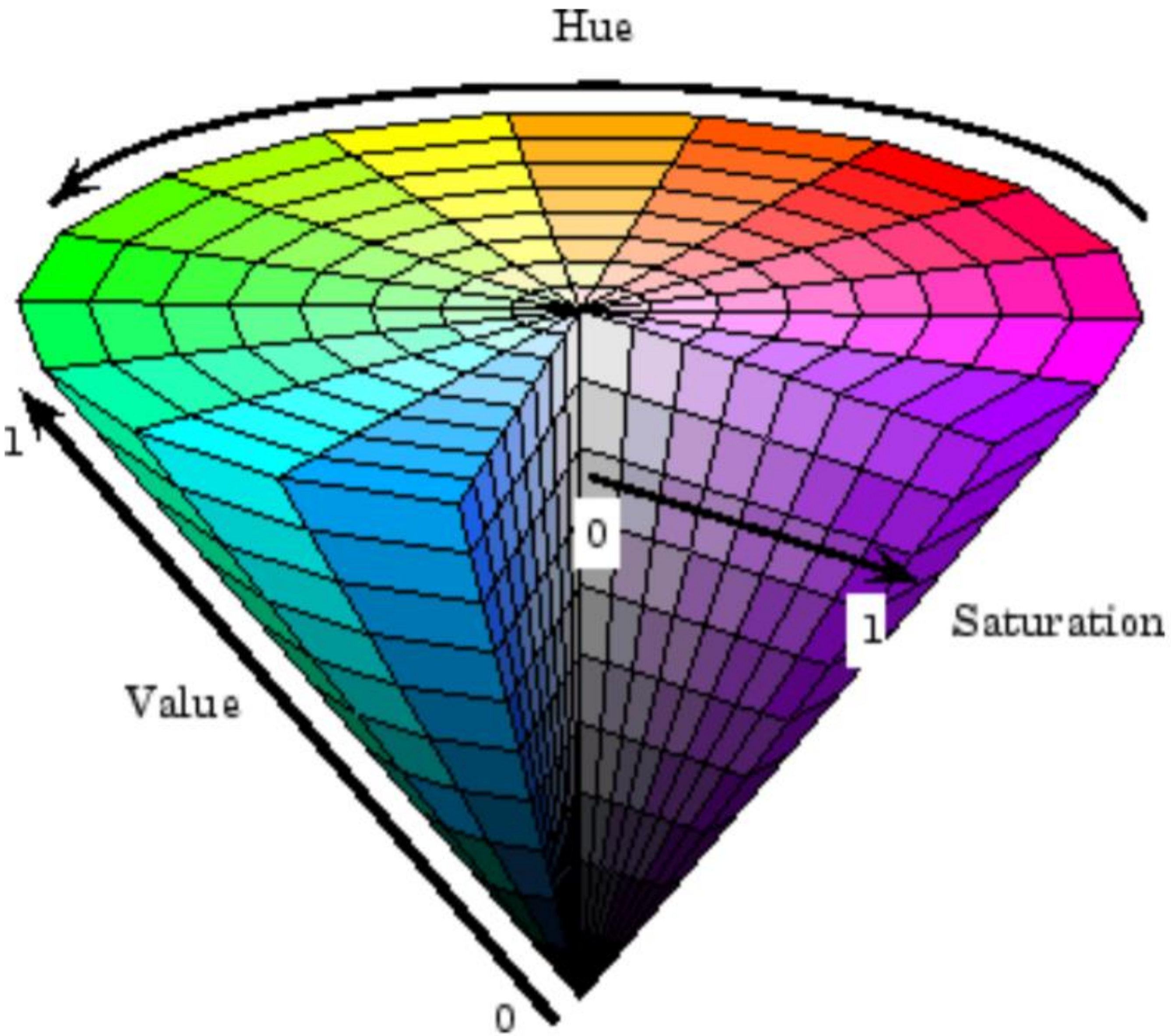
HSV

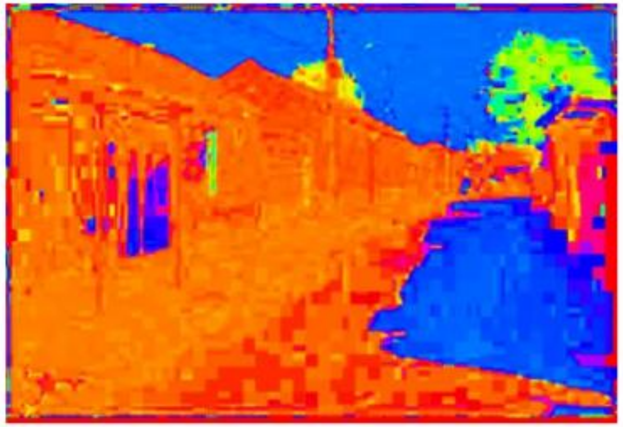
- **Hue**: The "attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors: red, yellow, green, and blue, or to a combination of two of them"
- **Saturation**: The "colorfulness of a stimulus relative to its own brightness"
- **Value**: The "brightness relative to the brightness of a similarly illuminated white". Can also be called **brightness or intensity**.
  - [Wikipedia]





Original image





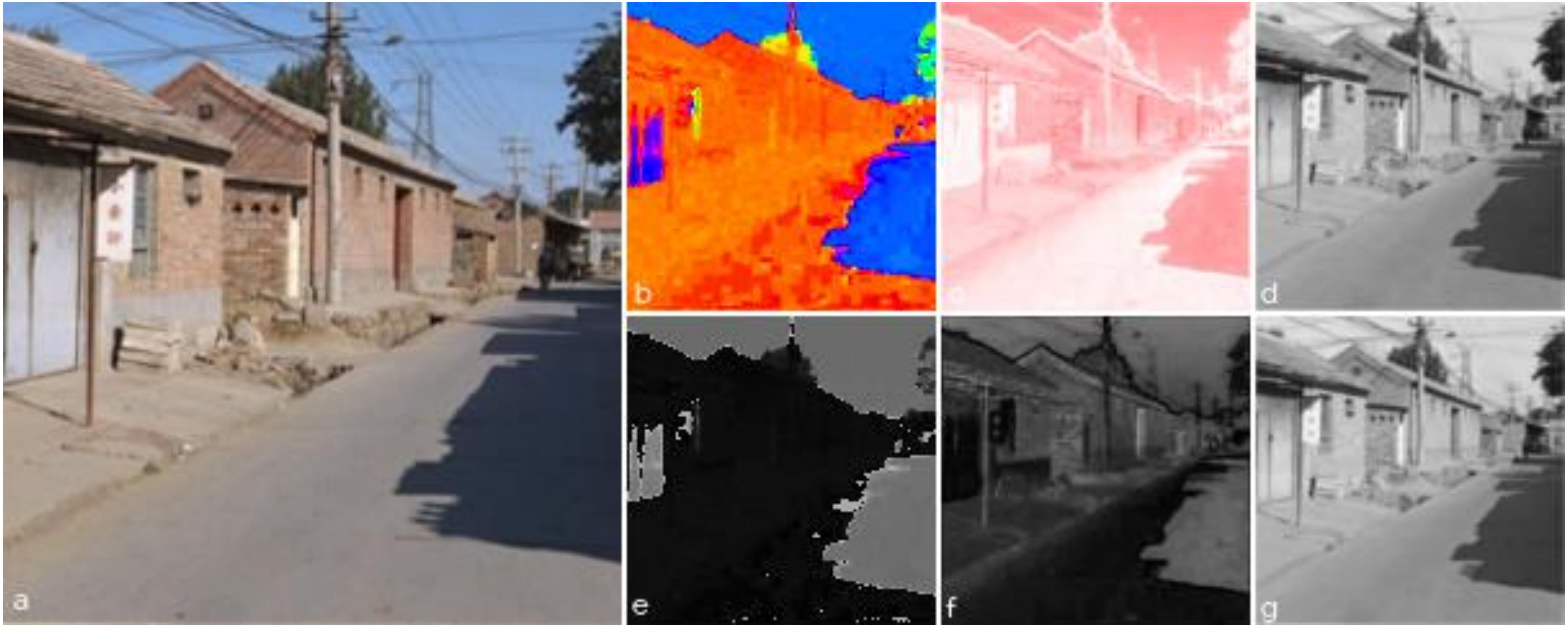
H
(S=1,V=1)

S
(H=1,V=1)

V
(H=1,S=0)

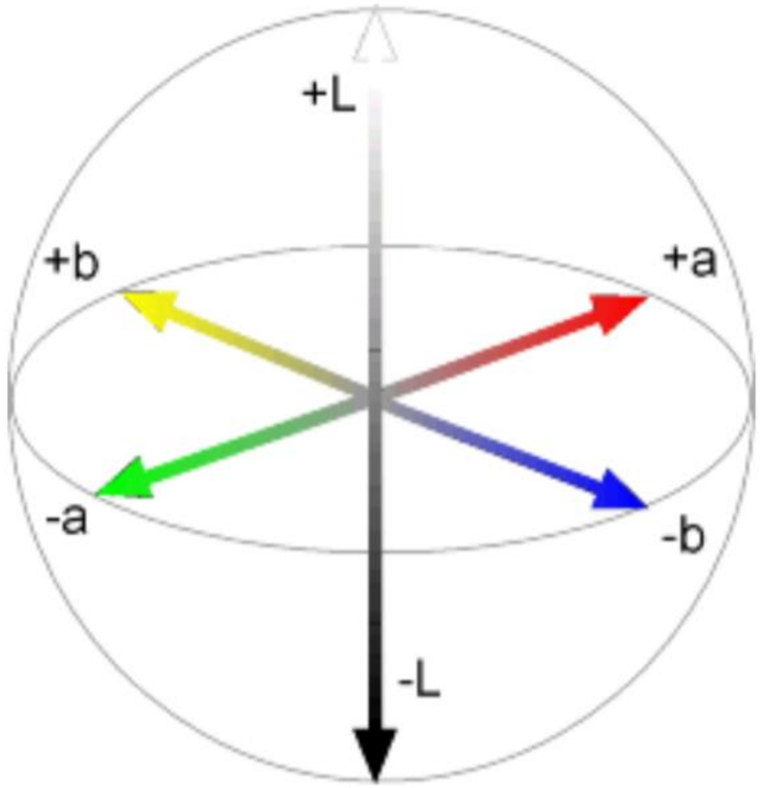
HSV

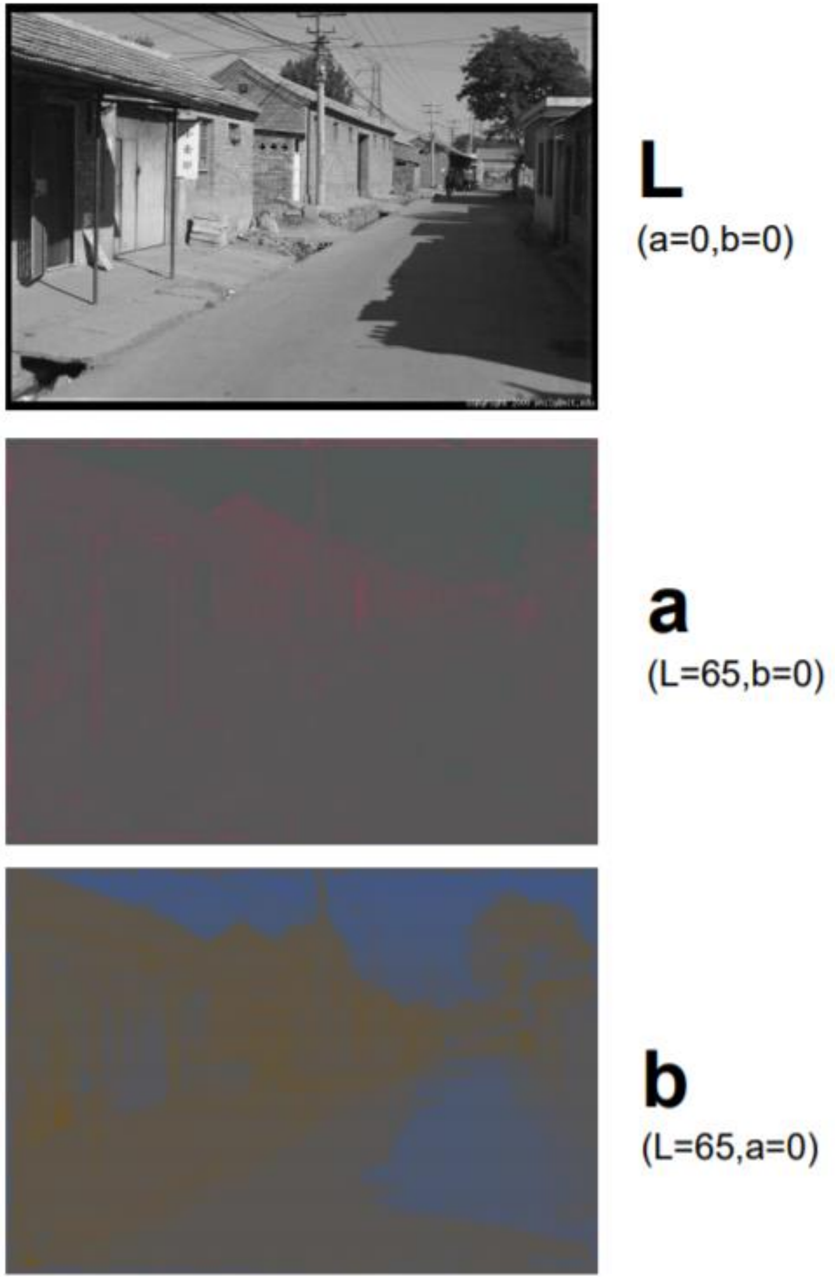
- In e, f, g: single channel image representation.
- Conclusion: people are much more responsive to intensity then chroma.



More color spaces: LAB

- L: lightness from black (0) to white (100).
- A: from green (−) to red (+).
- B: from blue (−) to yellow (+).





More color spaces: YUV

- Y: brightness/ intensity.
- U: blue projection.
- V: red projection.

- [Similar to YCbCr]