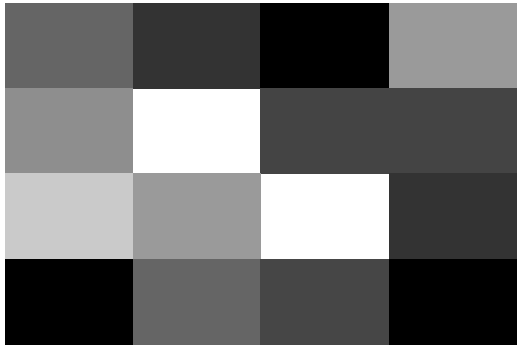
Digital Image Processing

Gaurav and Abhinav

What is a **digital** image ?

- An **array** of numbers representing color intensities

100	50	0	150
90	255	70	70
200	150	255	50
0	100	80	0

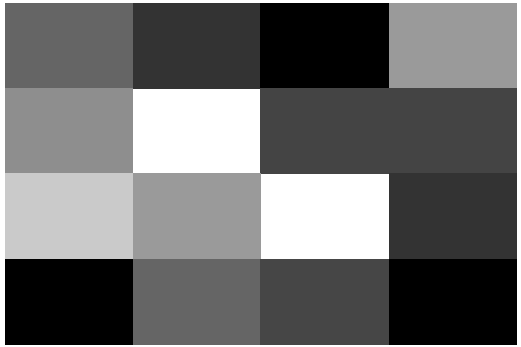


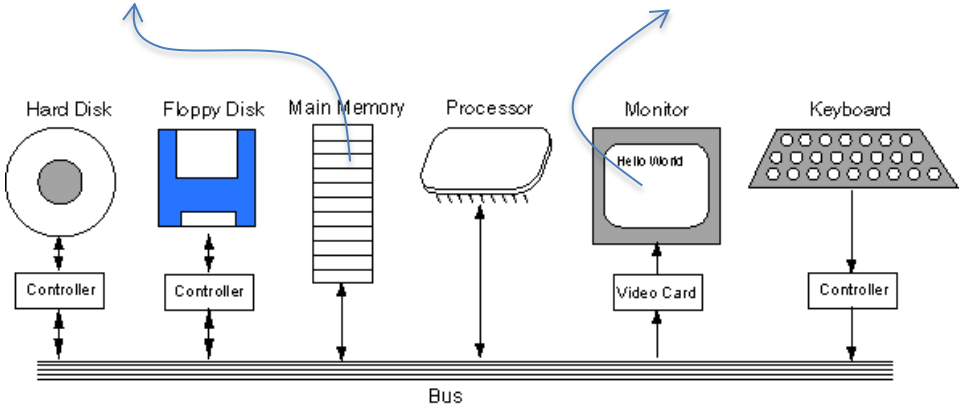
http://www.cs.ucc.ie/~gavin/cs1001/Notes/chap01/ch01_4.html

What is a **digital** image ?

- An **array** of numbers representing color intensities

100	50	0	150
90	255	70	70
200	150	255	50
0	100	80	0





http://www.cs.ucc.ie/~gavin/cs1001/Notes/chap01/ch01_4.html

What is a **digital** image ?

- An **array** of numbers representing color intensities

100	50	0	150
90	255	70	70
200	150	255	50
0	100	80	0

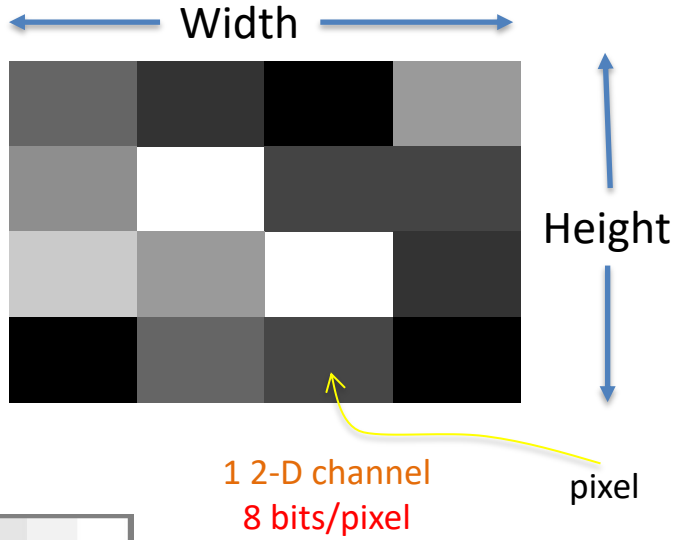
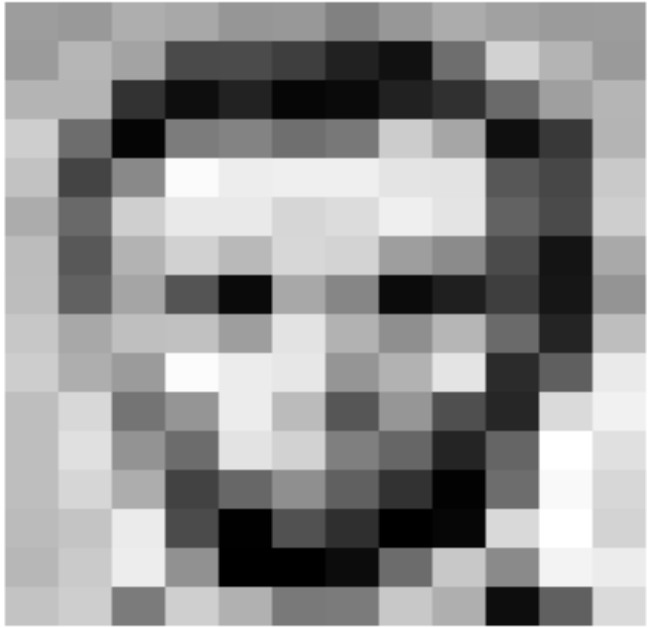


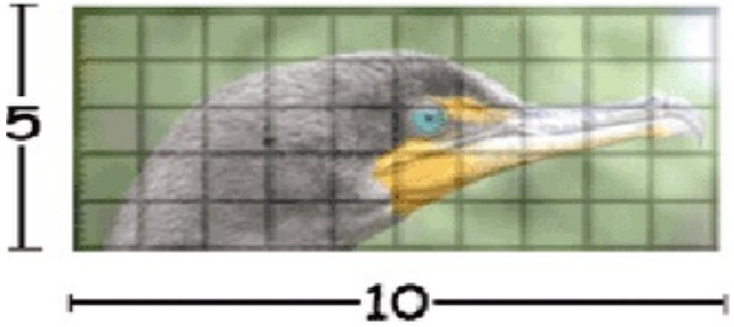
Image Representation (Grayscale)

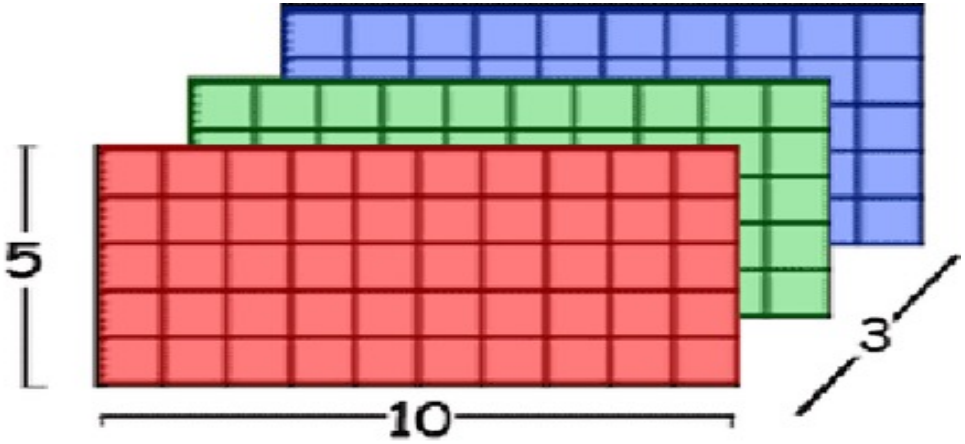


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

https://openframeworks.cc/ofBook/images/image_processing_computer_vision/

Image Representation (RGB)





3 2-D channels
24 bits per pixel

https://www.researchgate.net/profile/Bhupendra_Pratap_Singh/publication/282798184/figure/fig15/AS:283989639221249@1444719814399/Color-image-representation-and-RGB-matrix.png

Tasks of interest: Contrast adjustment

Image courtesy: mathworks

Tasks of interest: Edge Detection

Image courtesy: mathworks

Tasks of interest: Feature detection + stitching

Image courtesy: opencv

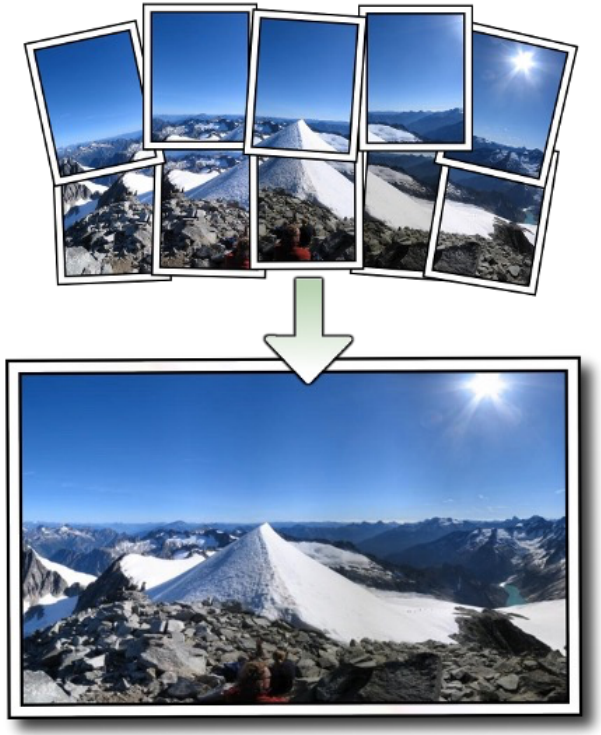


Image courtesy: autostitch

Tasks of interest: Noise Removal

Total variation denoising [Chambolle JMIV 2004]

Tasks of interest: Haze Removal

Single Image Haze Removal [He et al. CVPR 2009]

Retouch Personal Photos!

©Images taken from the web.

Tasks of interest: Artistic enhancement

Image courtesy: webneel.com

Image courtesy: Jon Morse

BW to Color

Mayabazar (1957), Vijaya Vauhini Studios

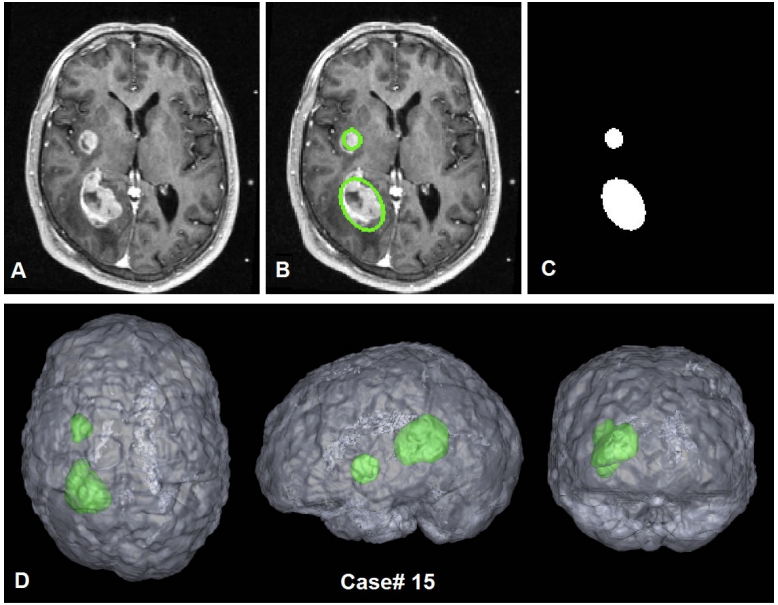
Colorized (2010)

Tasks of interest: Cinematic Grading

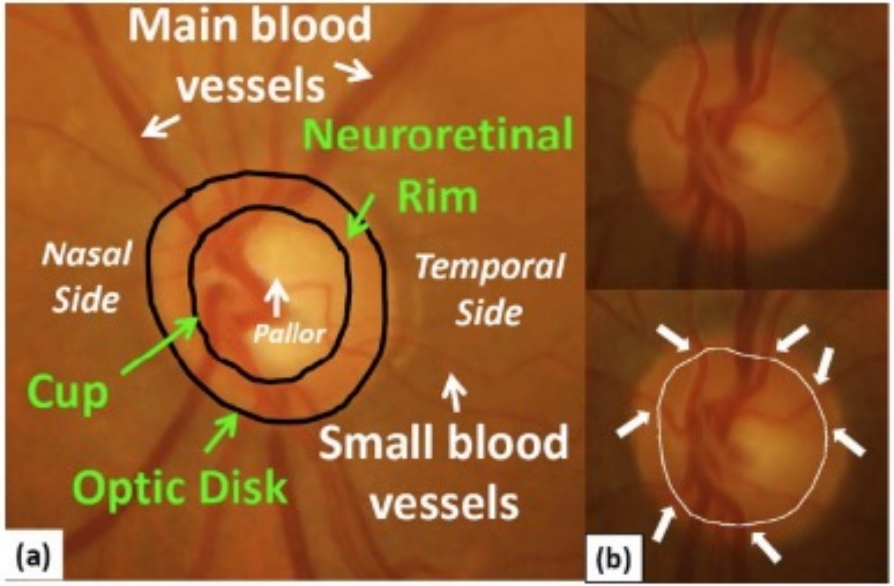
Mission Impossible - Ghost Protocol, Paramount Pictures

Image courtesy: juanmelara.com

Tasks of interest: Segmentation

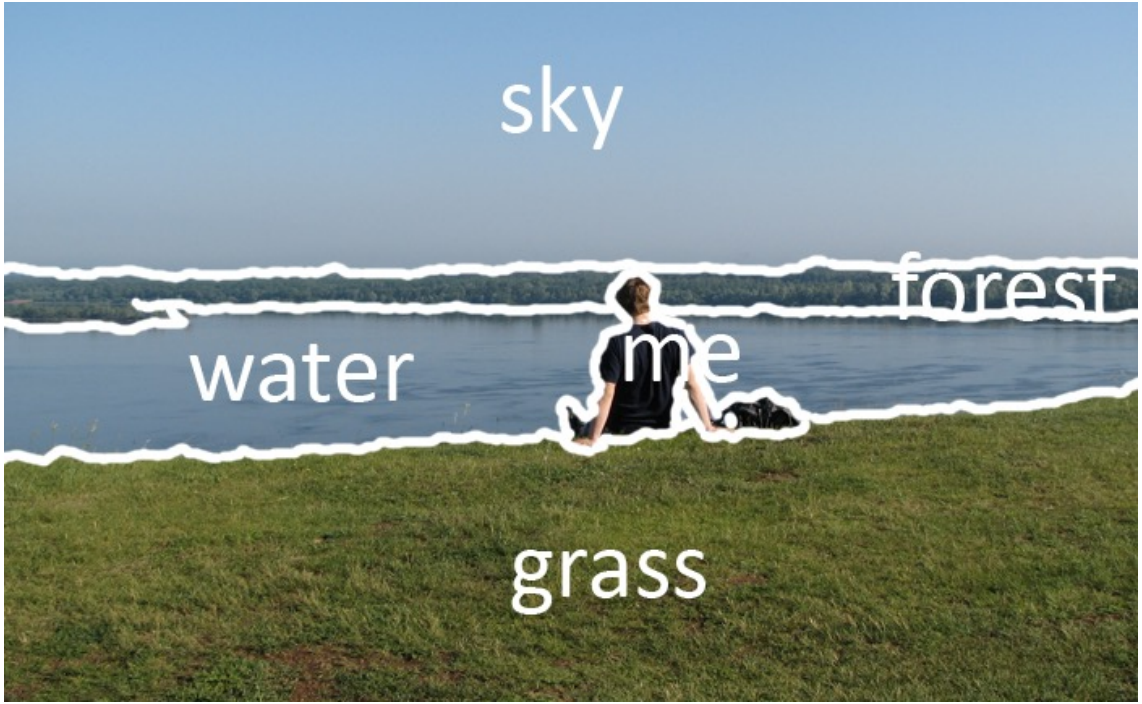


Tumour Segmentation [Yu et al. MICCAI 2010]



Cup Segmentation [Joshi and Sivaswamy 2011]

Tasks of interest: Segmentation



Courtesy: Roman Shapovalov

Tasks of interest: Compression

Original Image (1.2 mb)

Compressed JPEG Image (100 kb)

Tasks of interest: Inpainting

DAMAGED

RESTORED

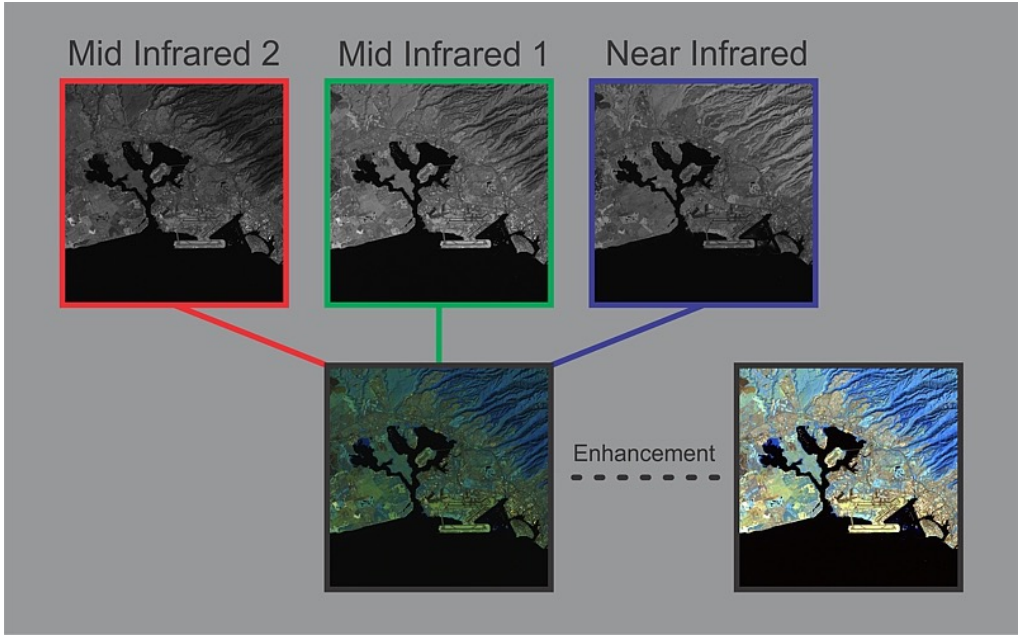
Bertalmio et al. SIGGRAPH 2010

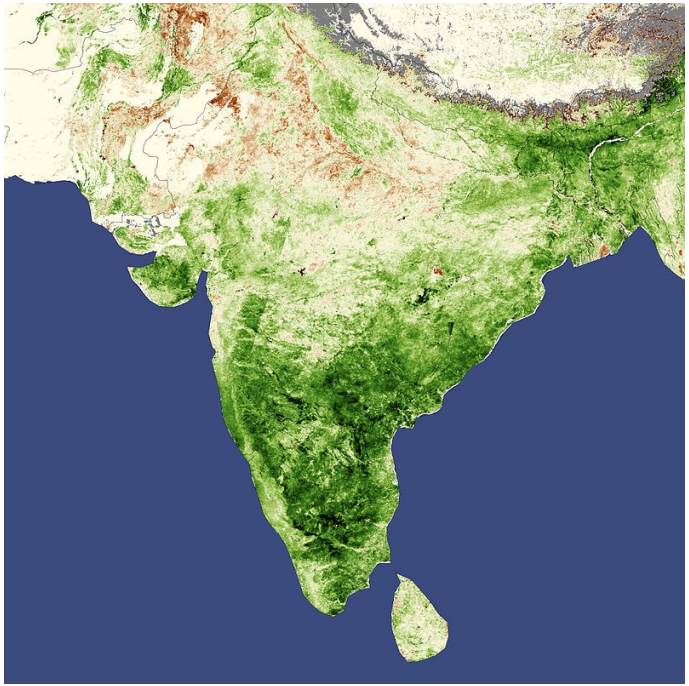
Tasks of interest: Special effects

courtesy: wachowski brothers (matrix)

courtesy: Miller et al. (sin city)

Tasks of interest: Satellite imaging

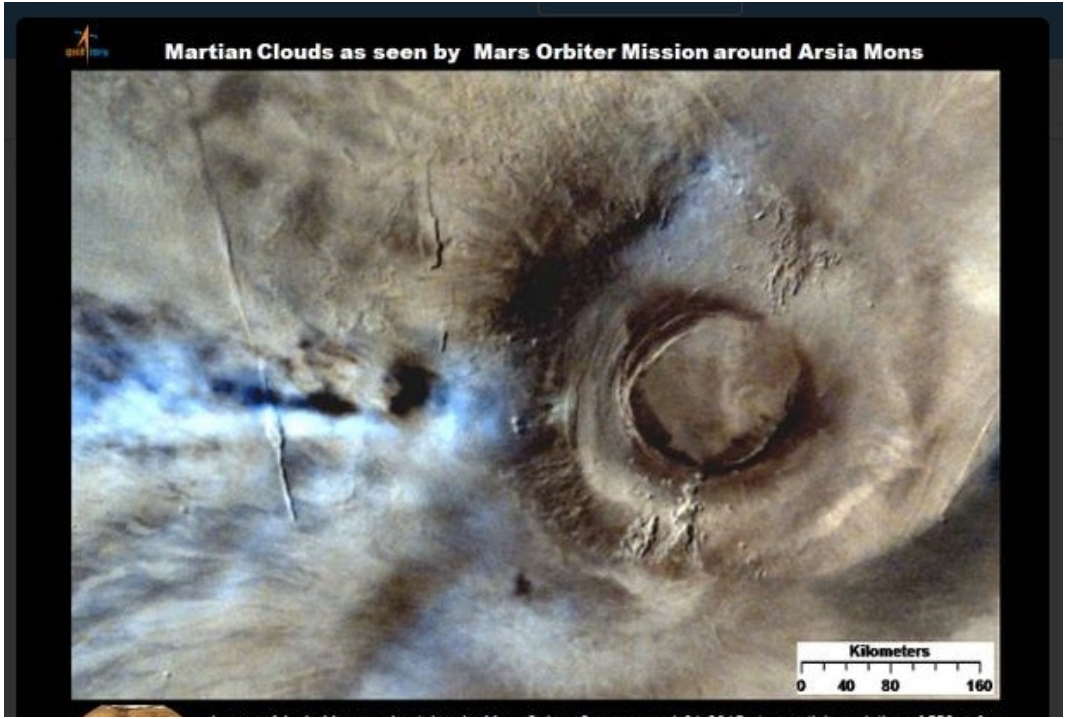


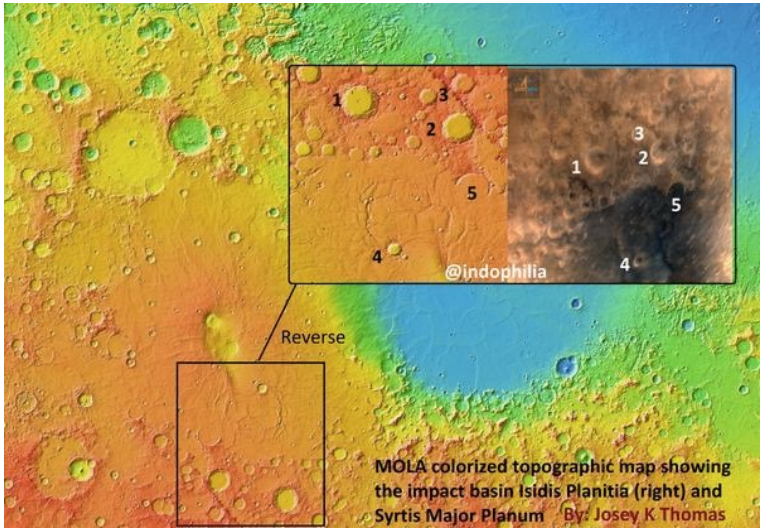


Terrain classification, weather predictions etc.

courtesy: NASA

Tasks of interest: Astronomy

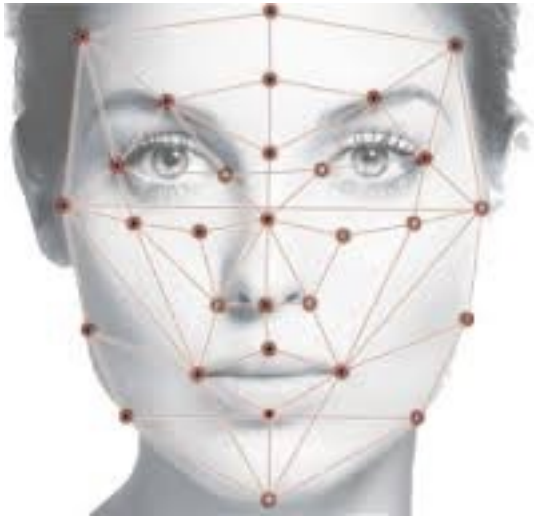




courtesy: ISRO

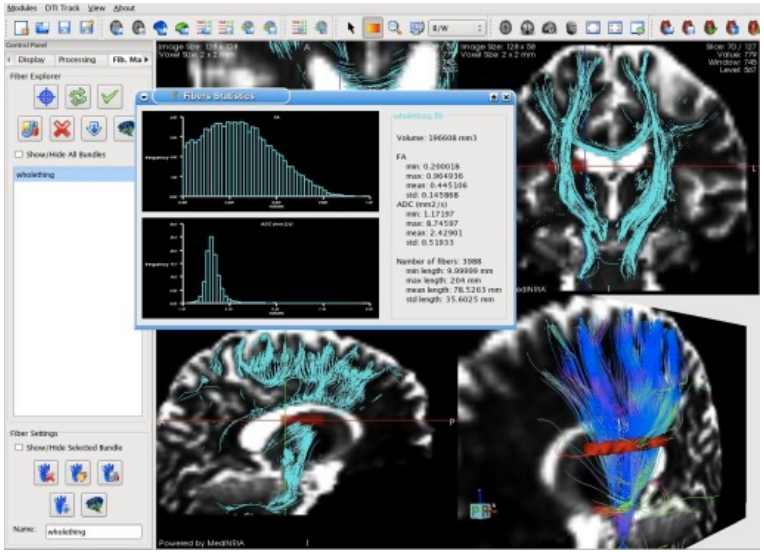
Tasks of interest: Biometrics

courtesy: dqindia.com

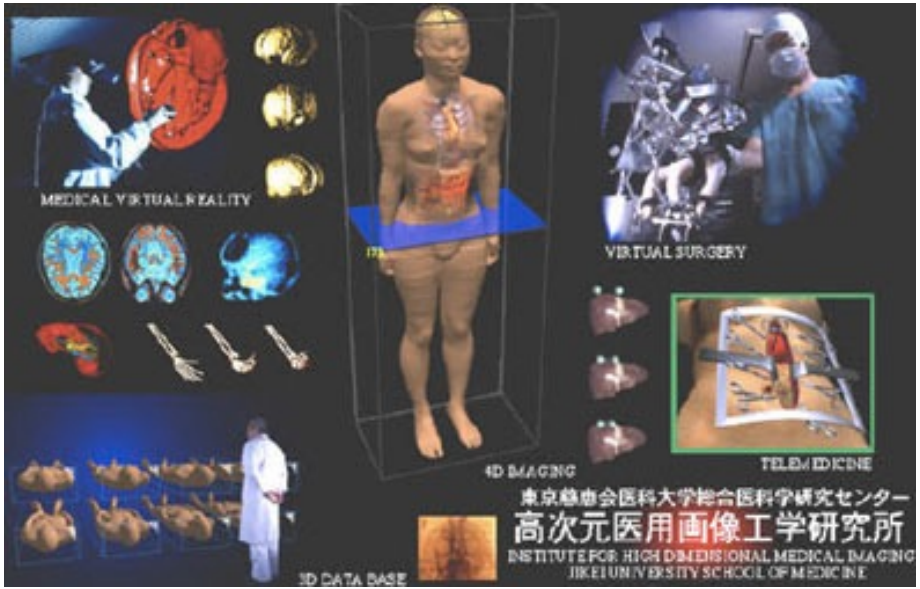


courtesy: heyce.com

Tasks of interest: Medicine



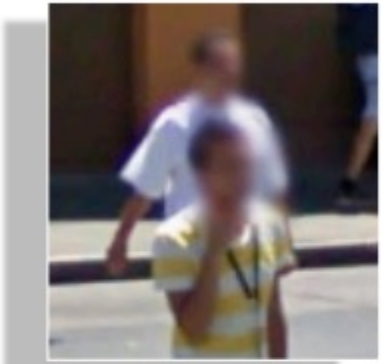
Courtesy: medINRIA



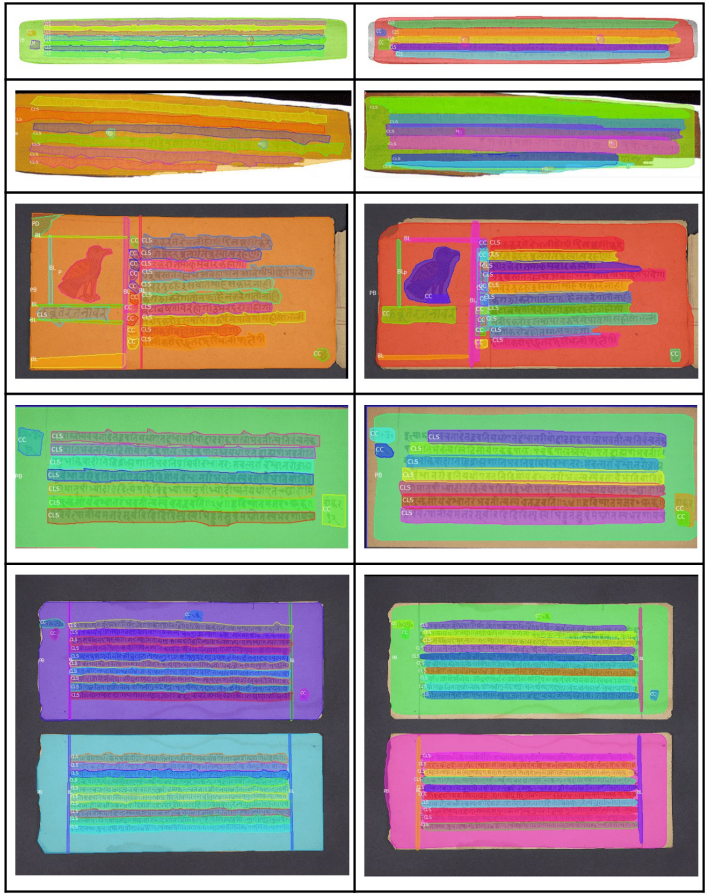
Courtesy: Naoki Suzuki

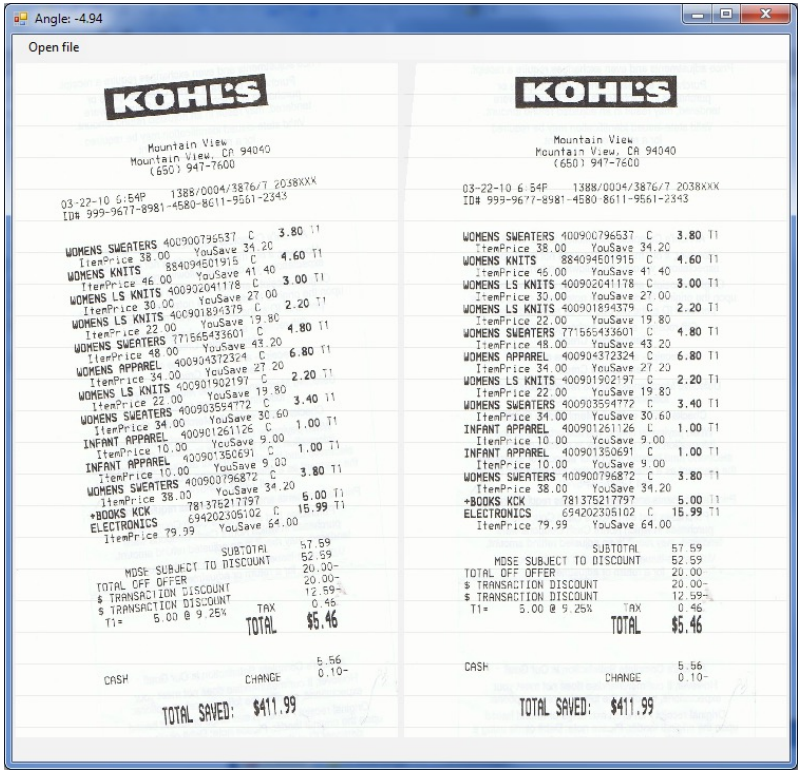
Tasks of interest: Driverless Vehicle Systems

Face Blurring for Privacy Protection



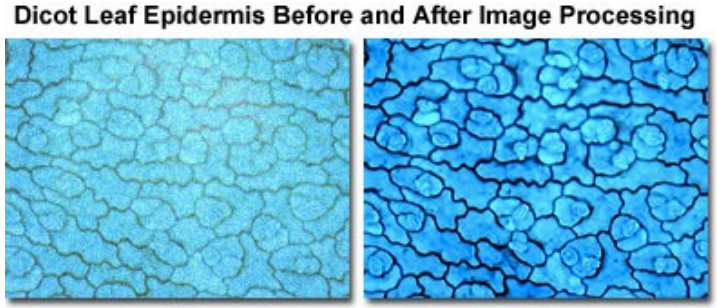
Tasks of interest: Document Image Analysis



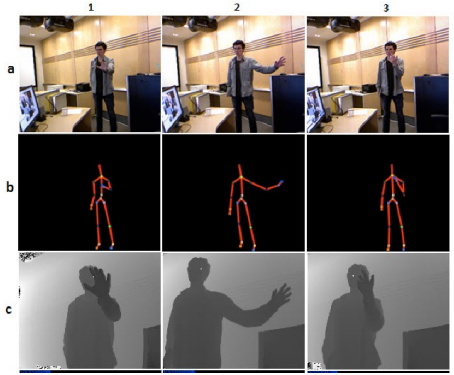


Tasks of interest: Many more

- Biology
- HCI
- Number Plate recognition
- Gesture recognition



Courtesy: Olympus



Courtesy: Perviverzov et al. 2012

Courtesy: researchdesignlab.com

Two Paradigms of Image Processing

Directly in Spatial Domain

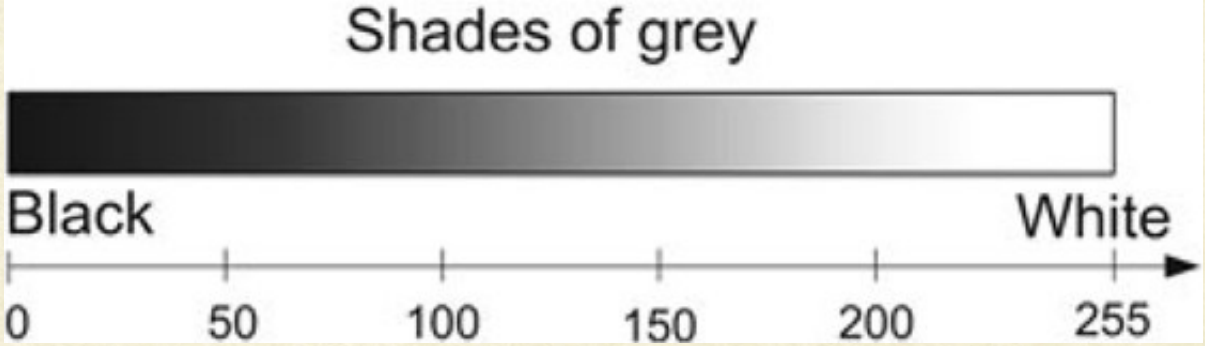
- Manipulating Pixels
  - Easier to Understand
  - Easier to Implement
  - Local Changes
  - Not so efficient

Processing in Transform Domain

- Manipulating JPG coefficients
  - Complex to visualize
  - Tricky to implement
  - Changes can be global
  - Can be very efficient, effective, …

Linear Intensity Transforms

- $T(z) = z + K$
- $T(z) = z - K$
- $T(z) = Kz$
- $T(z) = K_1 z + K_2$



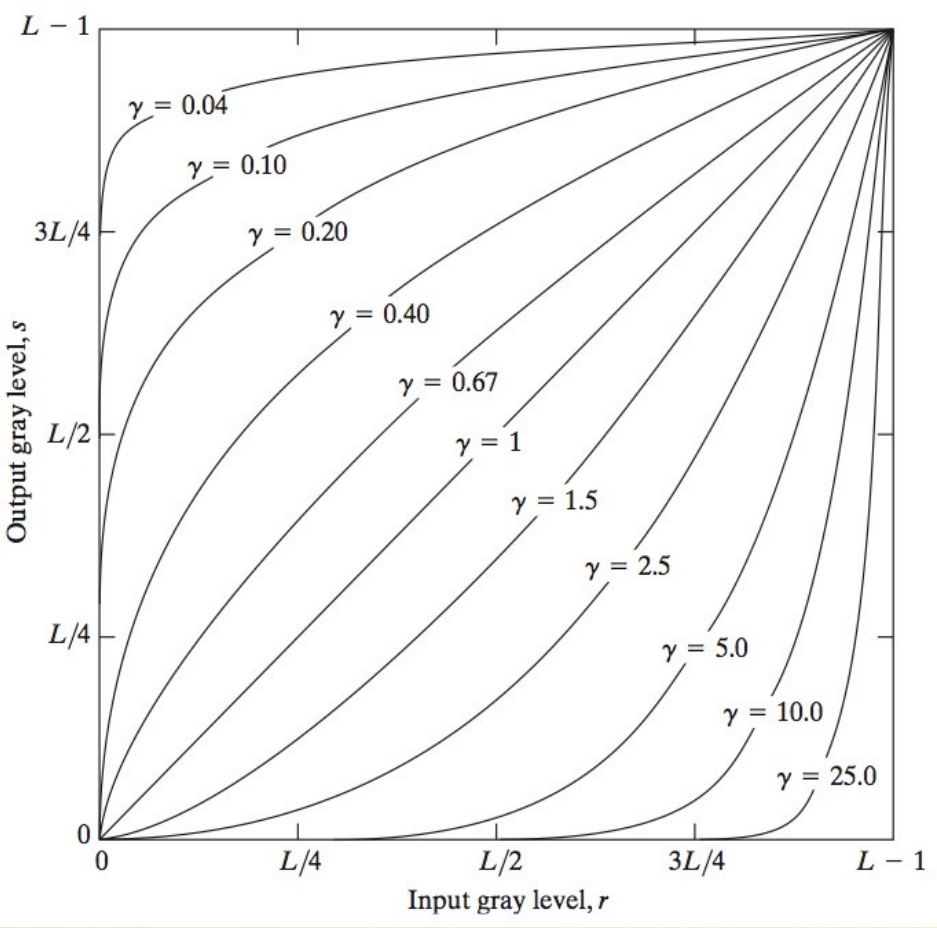
e.g.,

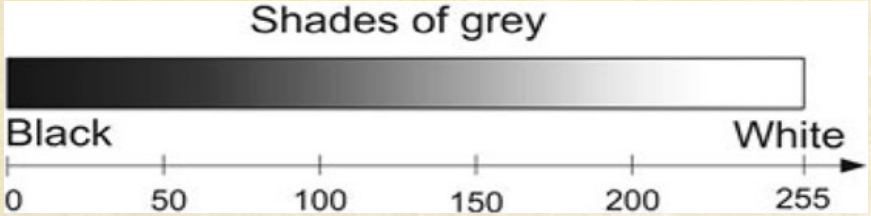
- Image Negative
- Data Visualization:
  - Map to display range

$$J = \mathtt{round}\left(255 * \frac{I - min(I)}{max(I) - min(I)}\right)$$

Power-Law Transformations

$$s = cr^{\gamma}$$





a b
c d

FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0$, and 5.0, respectively. (Original image for this example courtesy of NASA.)

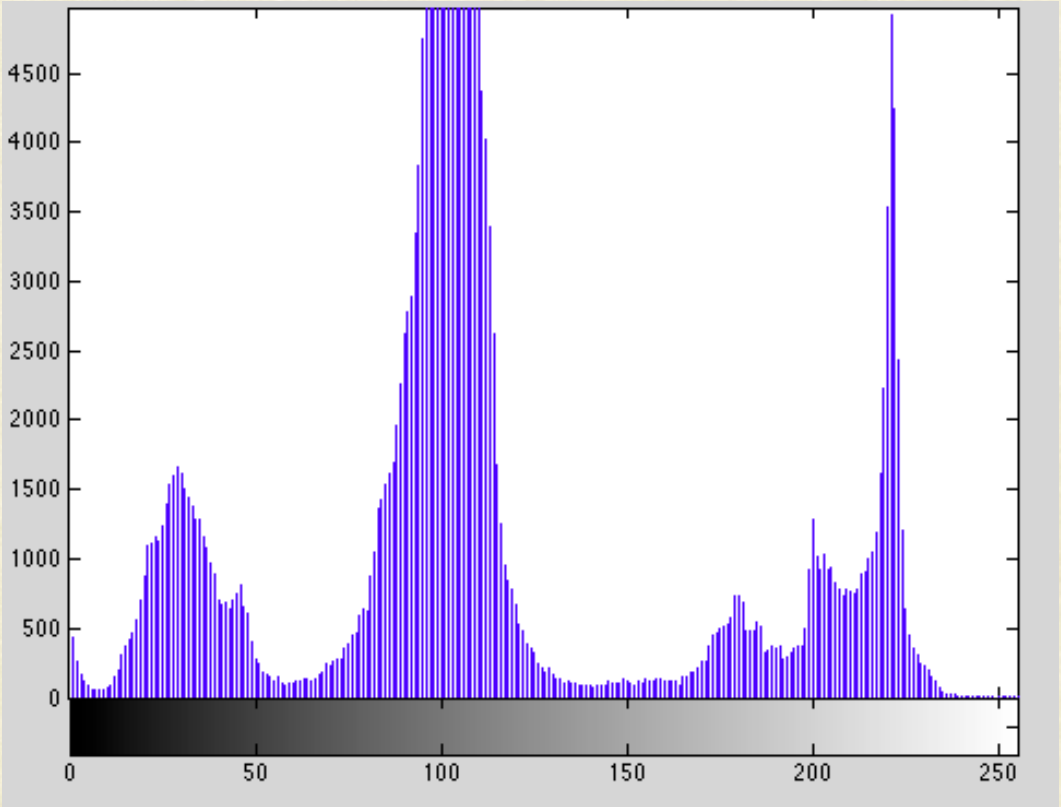
Demo:
https://colab.research.google.com/drive/11qIL0VKleZnONtPuxAryAf9WkUC7kEMI#scrollTo=aU5WQagOpSCr&line=12&uniqifier=1

Histogram

$$h_r(i) = n_i$$

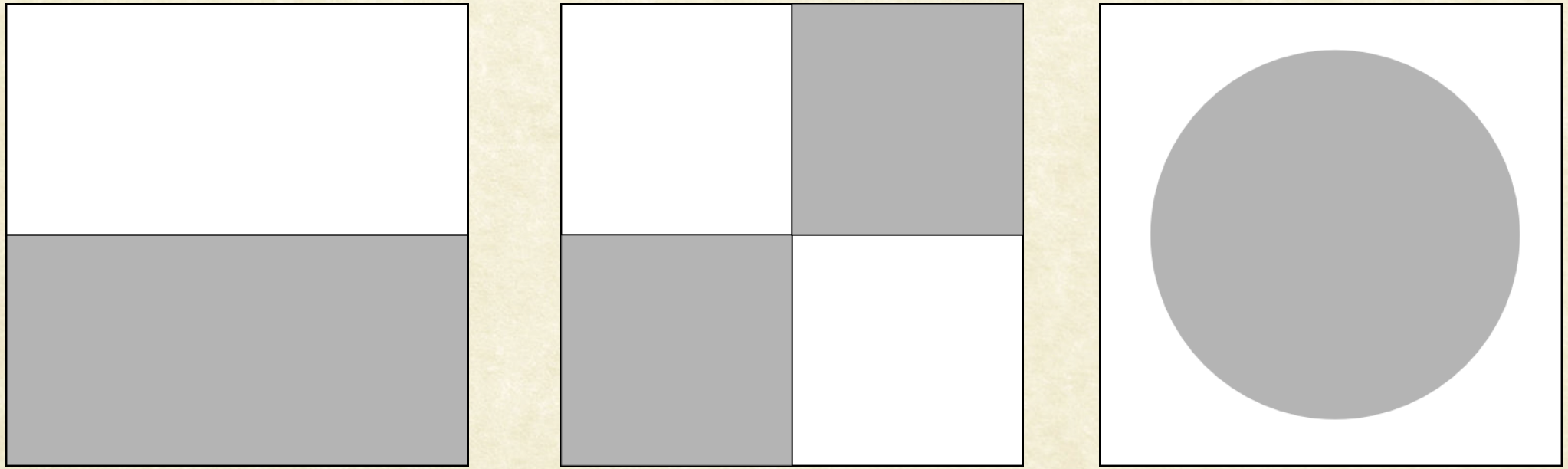
i → intensity value, range [0,L-1]

n_i → number of pixels with intensity i



Histograms

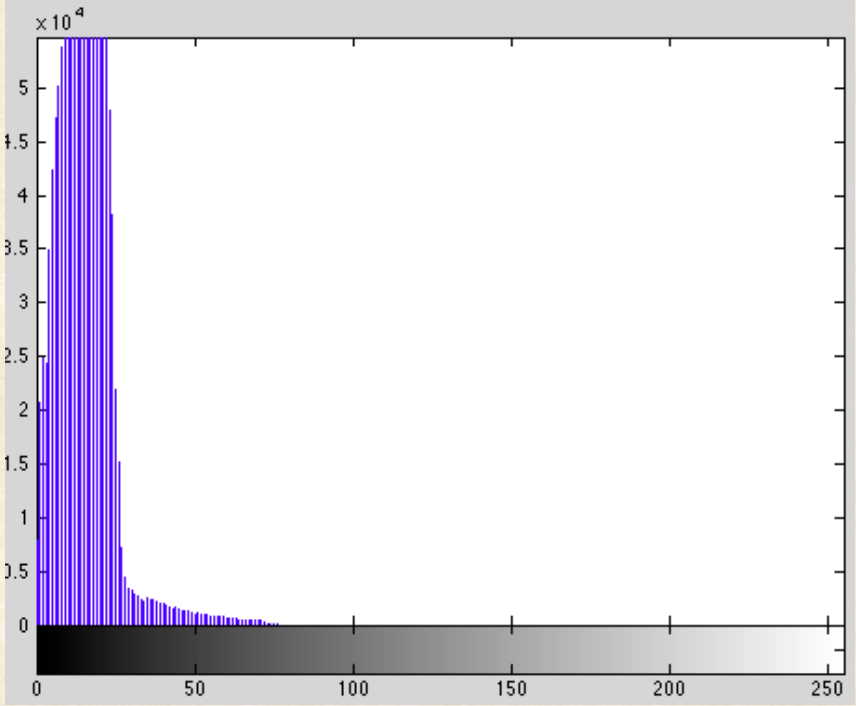
- Different images can have same histogram



- No information about spatial distribution of intensity values

Histograms

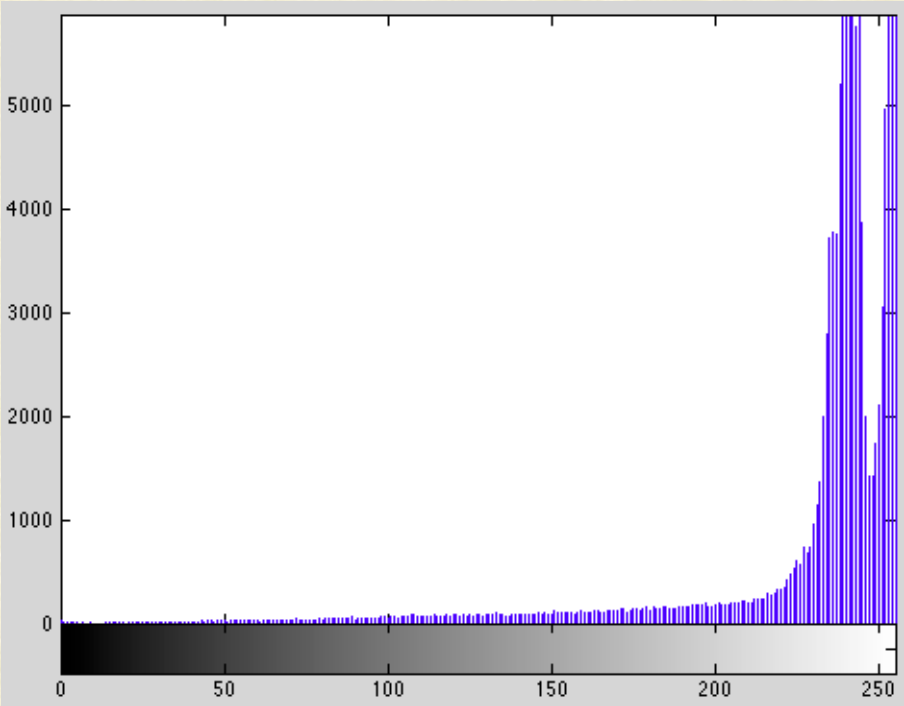
- Histograms and brightness



Under exposure

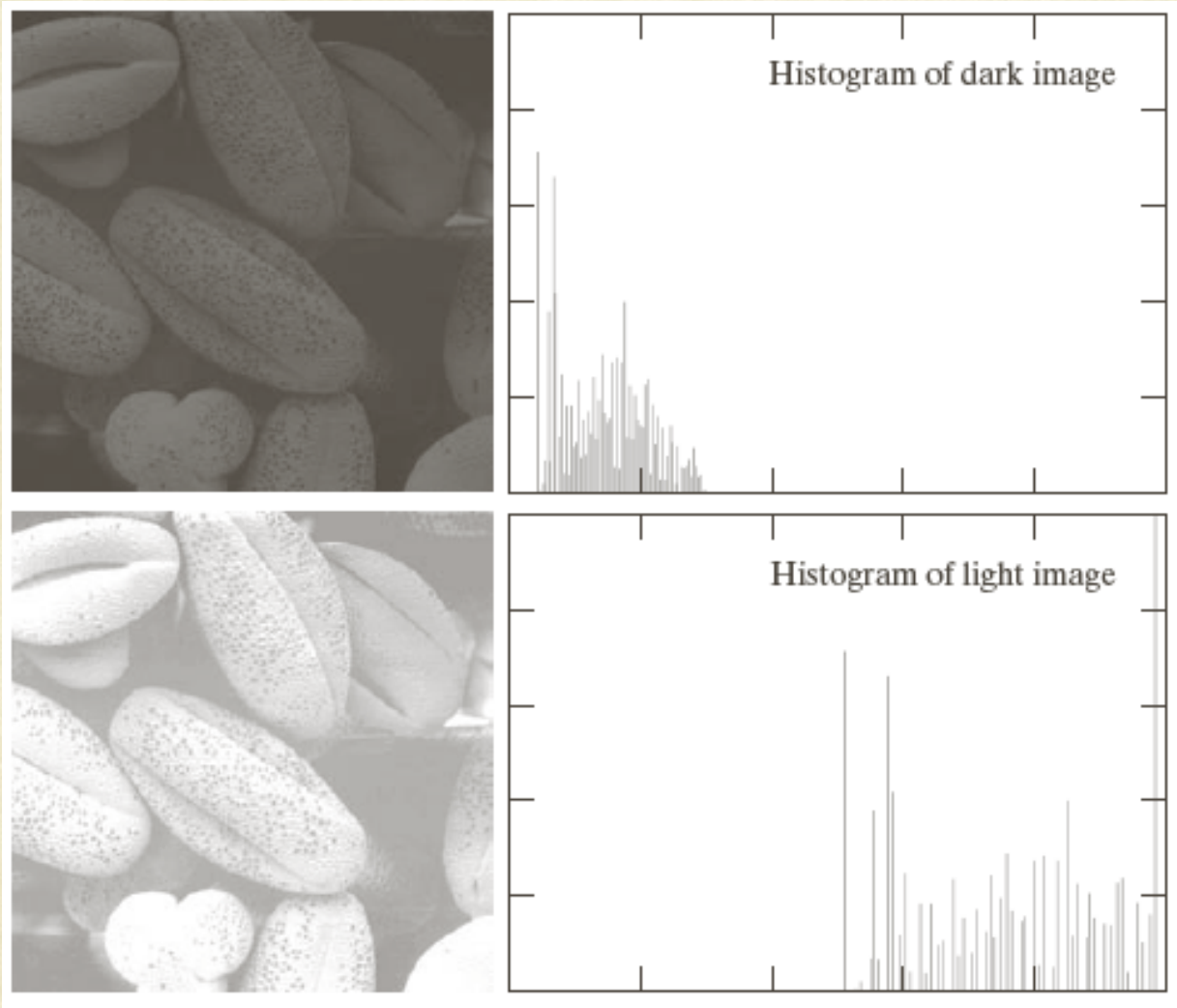
Histograms

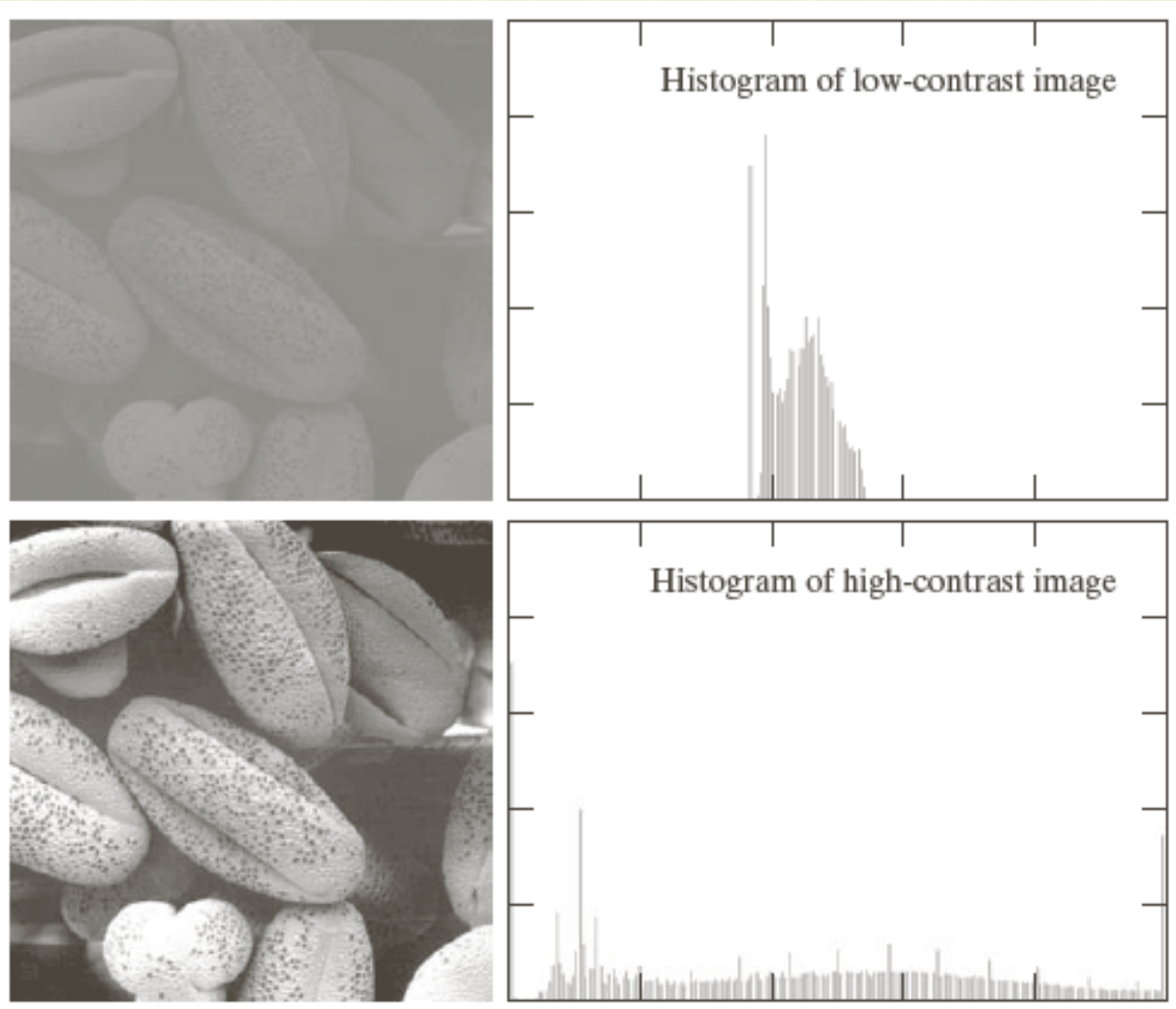
- Histograms and brightness



Over exposure

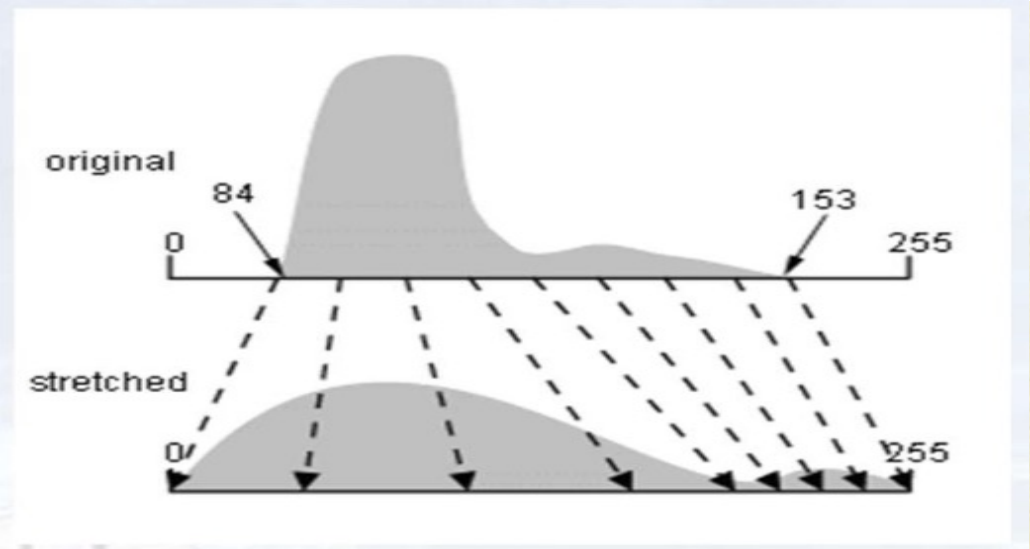
Histogram and Contrast

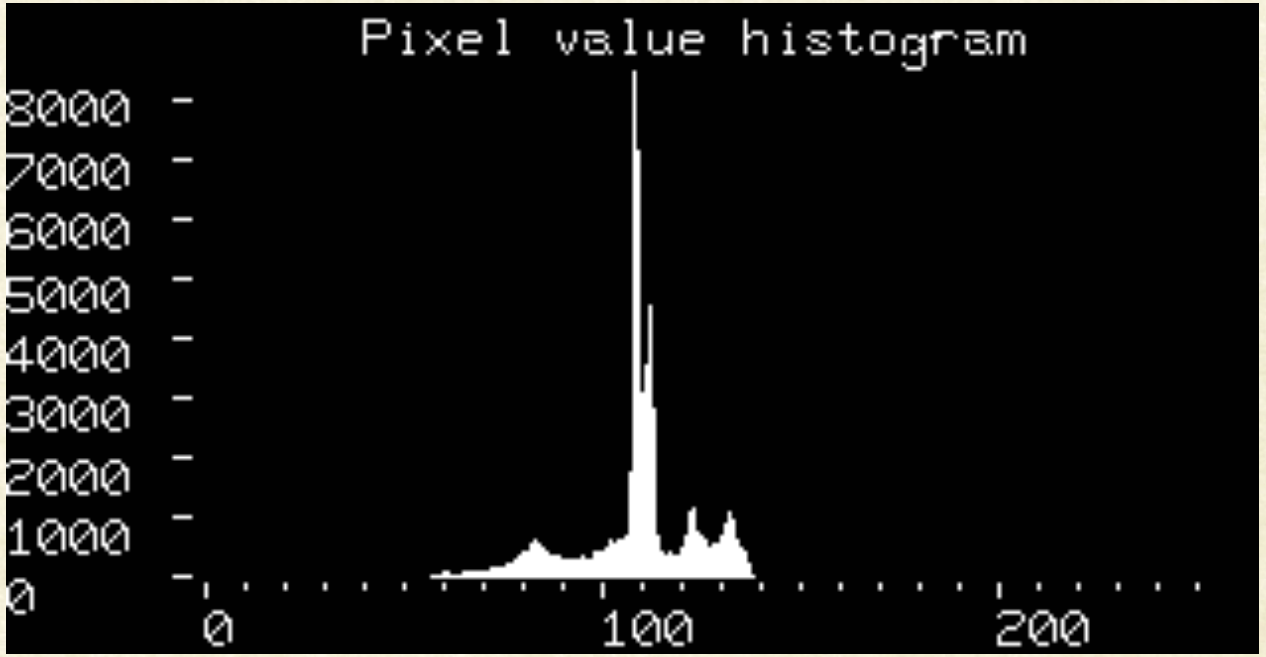


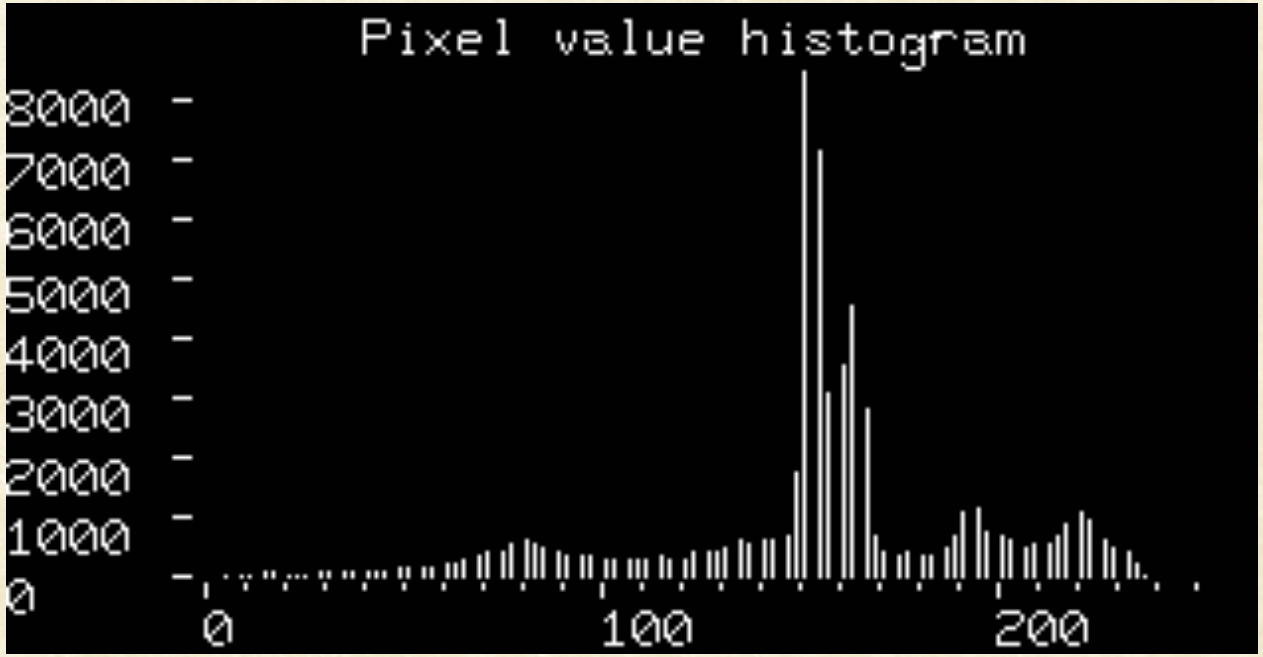


Contrast Stretching

$$f_{ac}(a) = a_{min} + (a - a_{low}) \cdot \frac{a_{max} - a_{min}}{a_{high} - a_{low}}$$

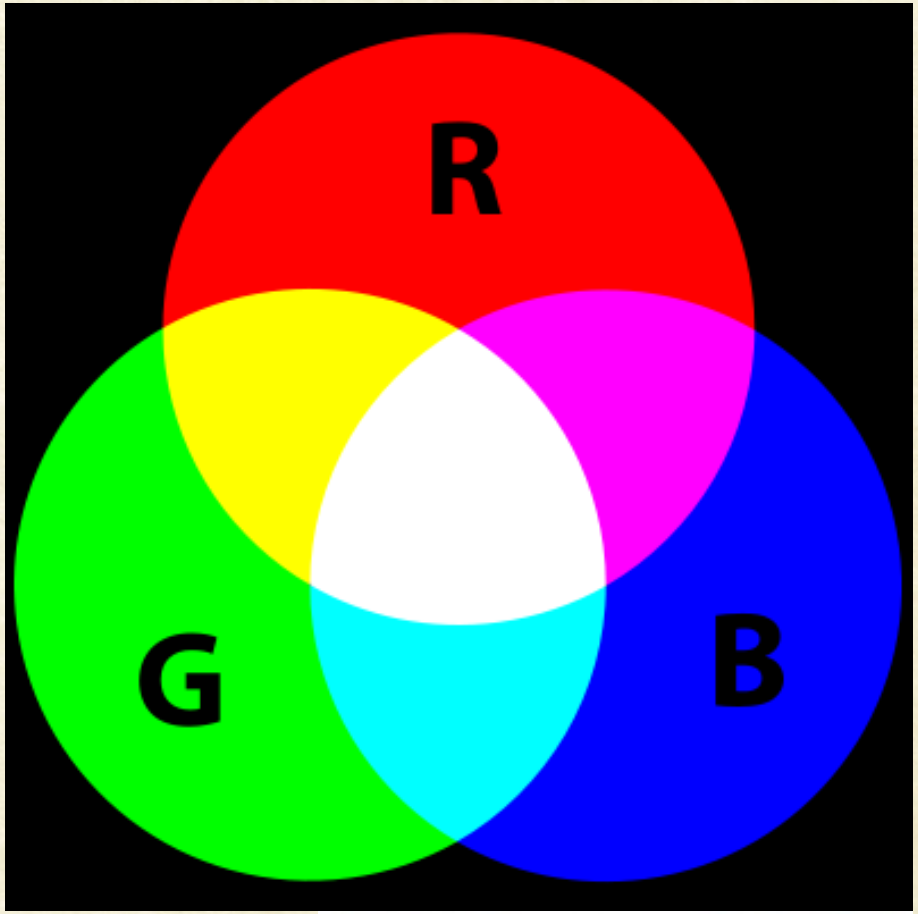



Pixel value histogram
8000
7000
6000
5000
4000
3000
2000
1000
0
0
100
200


Pixel value histogram
8000
7000
6000
5000
4000
3000
2000
1000
0
0
100
200

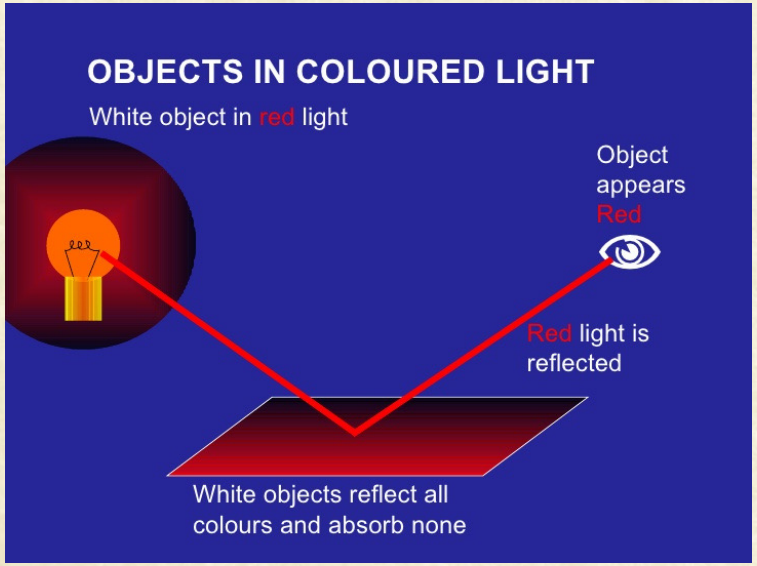
Primary Colors (color as three numbers)

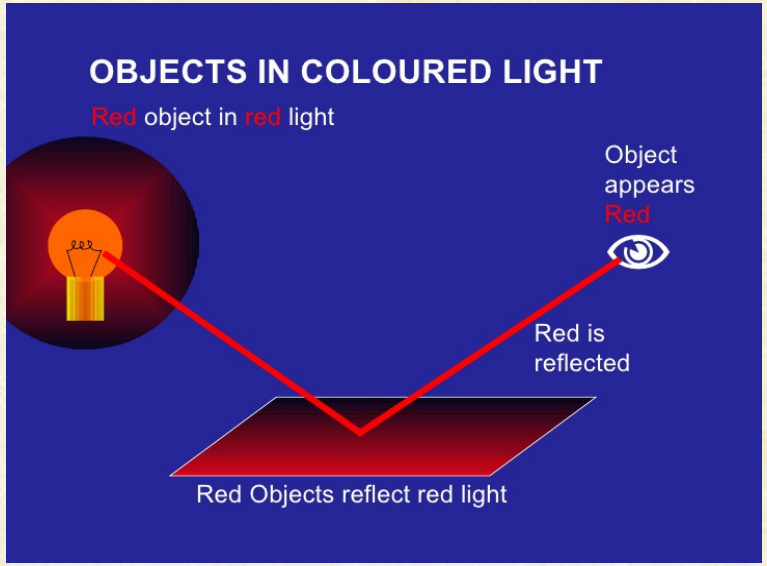
- Additive (CRT displays, projectors etc.)
  - Combining red, green and blue light produces lighter colors, offering a good contrast to dark screens.
  - (Usually) Combined on a surface which reflects all light that falls on it
  - Easy to get **black** – don't shine any light (CRT)

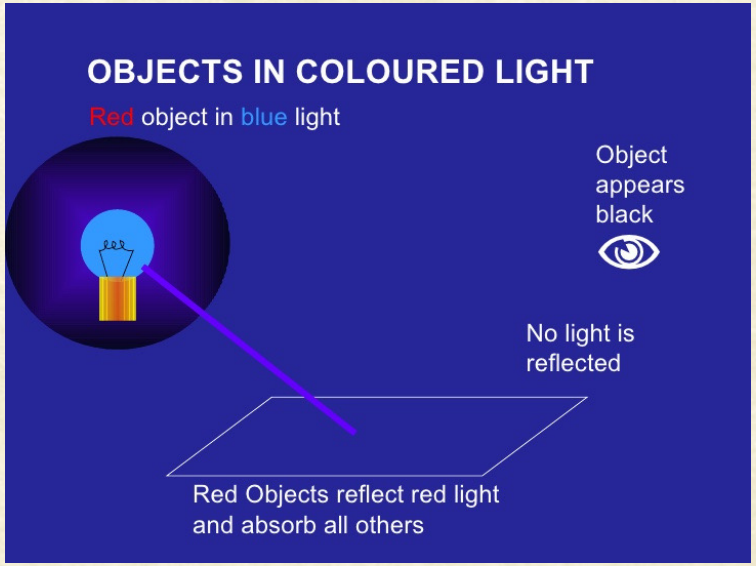


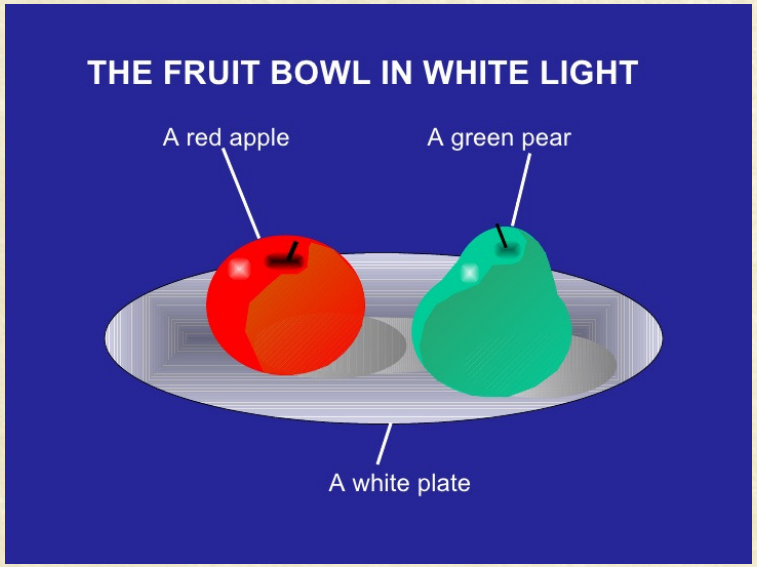
Magenta = Red + Blue
Cyan = Blue + Green
Yellow = Green + Red

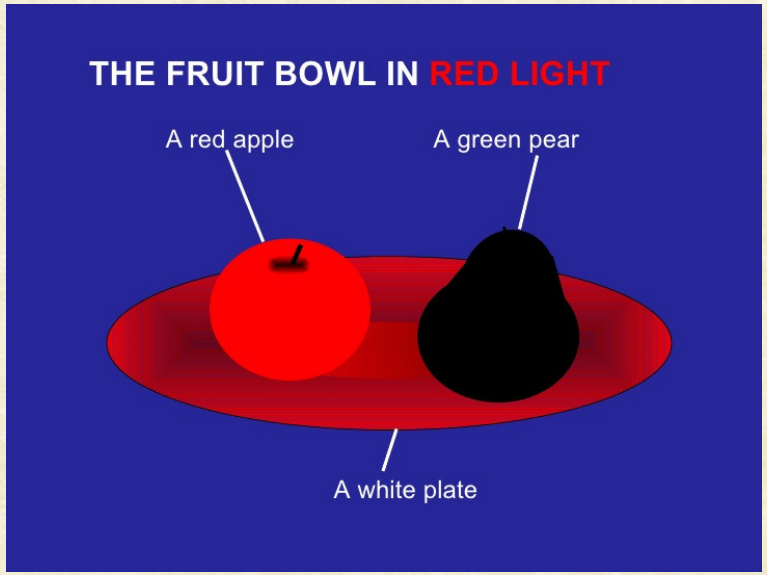
Perceived Color = f(Light, Object)

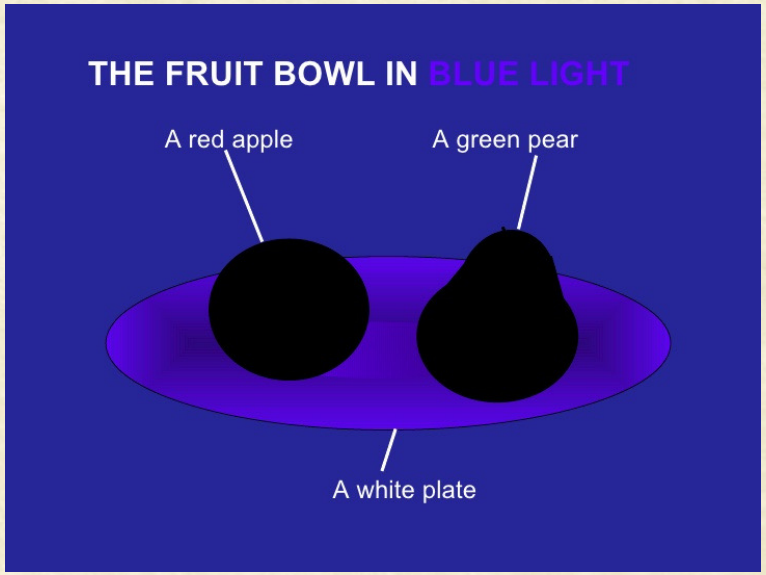












R,G,B Channels of an Image

Full color

Red

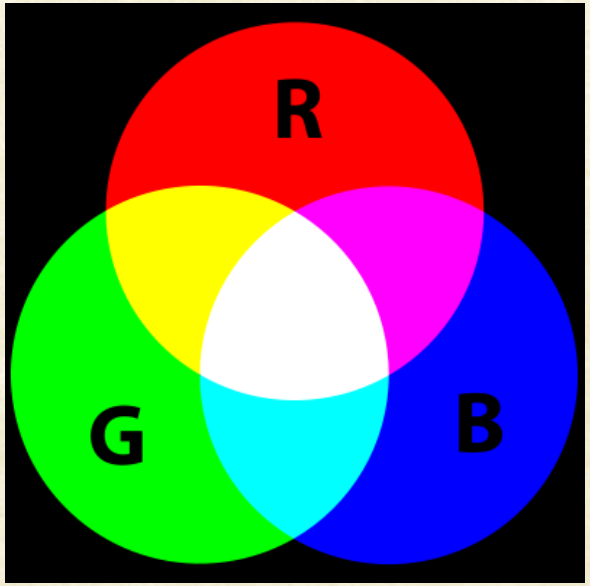
Green

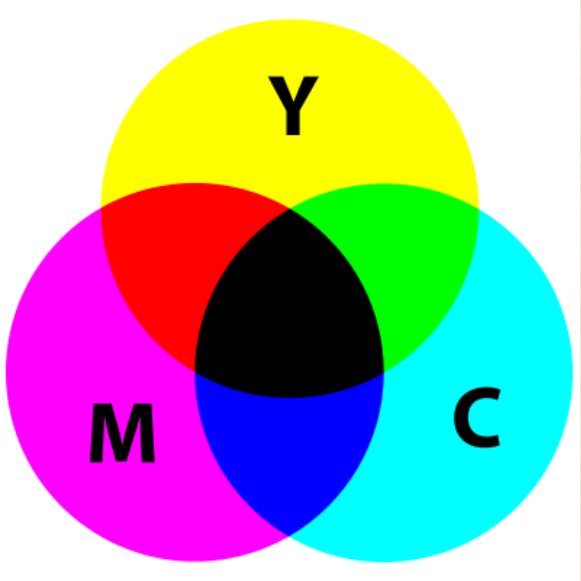
Blue

R,G,B Channels of an Image

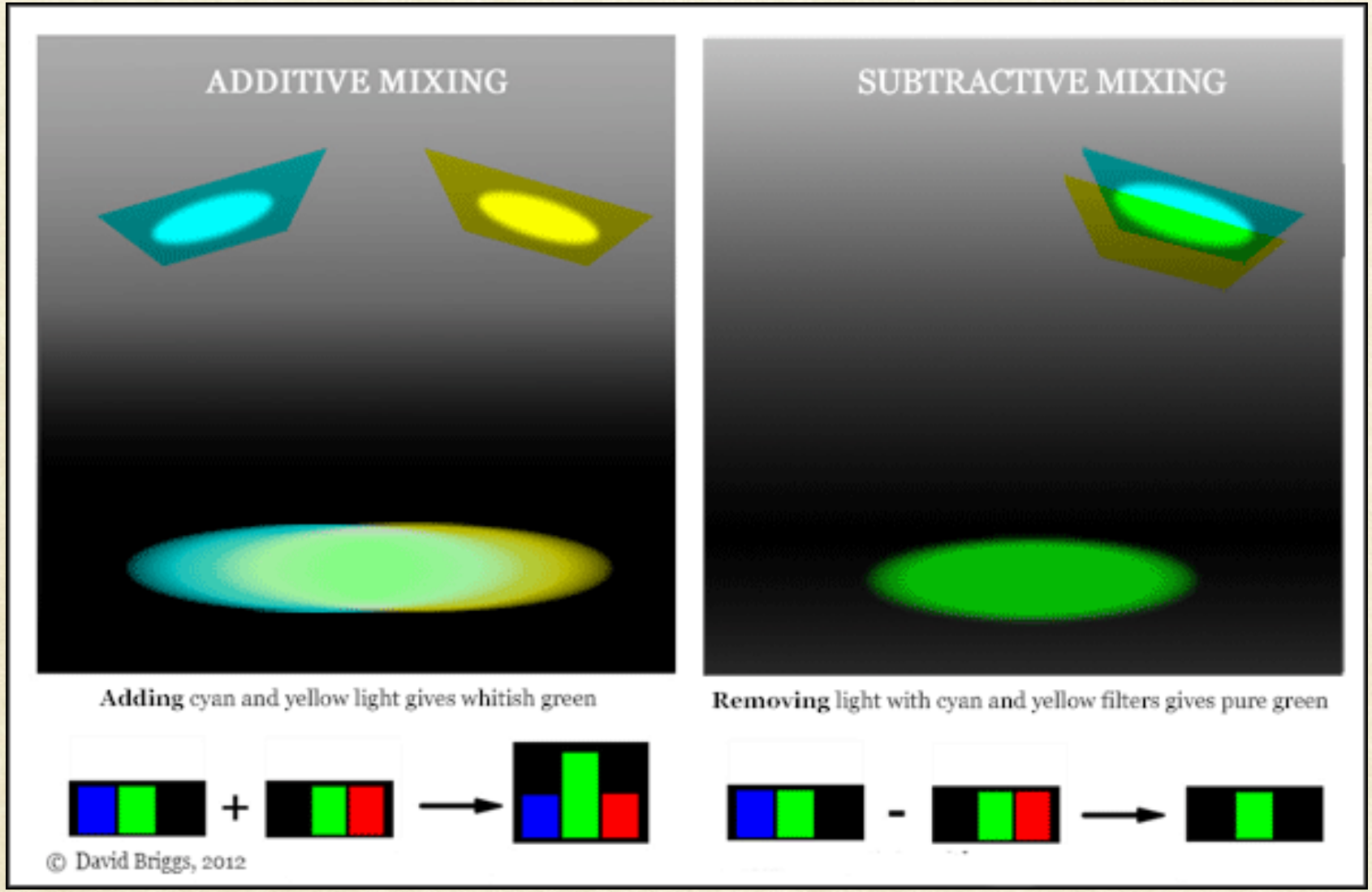
Primary Colors

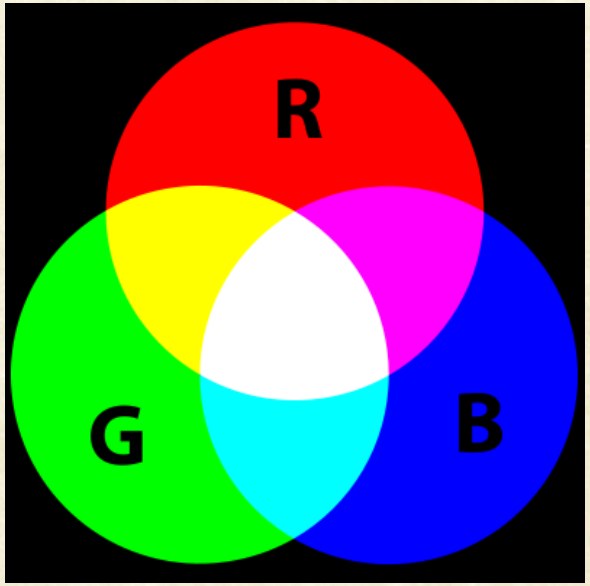
- Subtractive (Painting, printing, etc.)
  - Prevent certain wavelengths from being reflected
  - Starting with white paper, each added colored ink "subtracts" available color from the starting medium.
  - For pure white , leave the paper unprinted (nothing can be whiter than the paper it is printed upon).

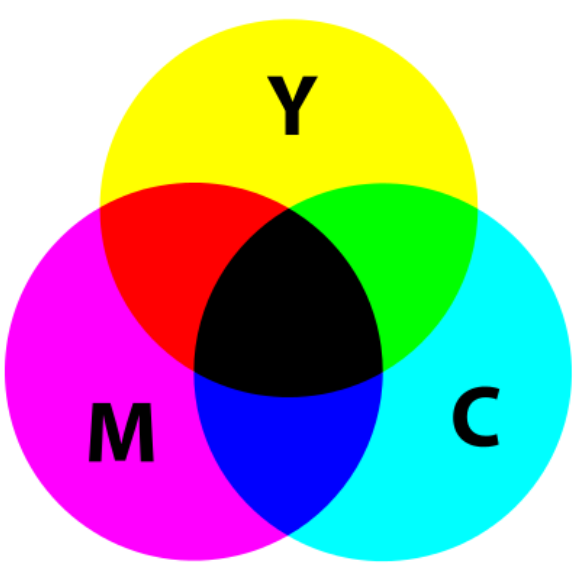




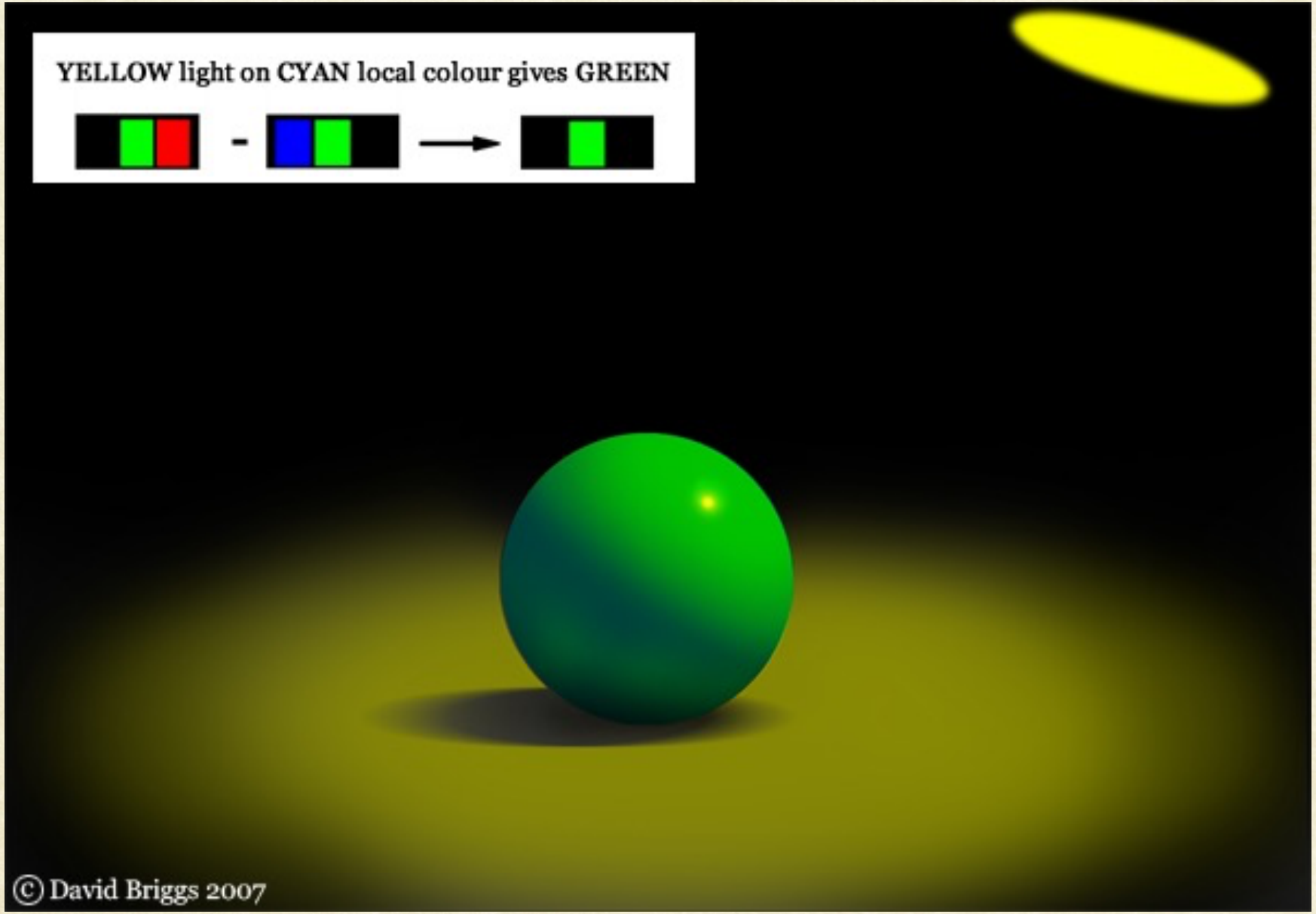
RGB vs CMY


ADDITIVE MIXING
SUBTRACTIVE MIXING
Adding cyan and yellow light gives whitish green
Removing light with cyan and yellow filters gives pure green
© David Briggs, 2012

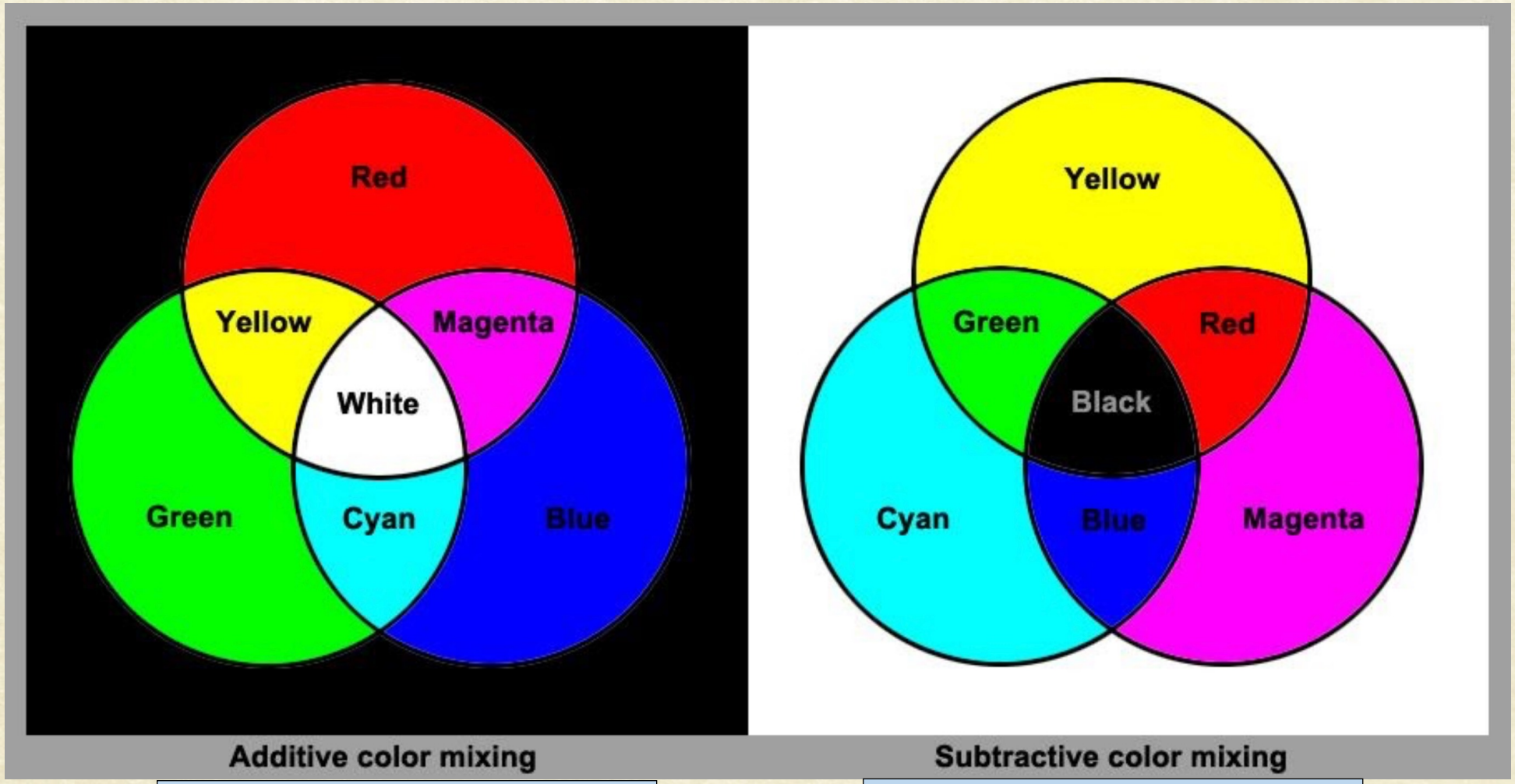

R
G
B


Y
M
C

Objects in Colored Light



Additive vs. Subtractive color mixing

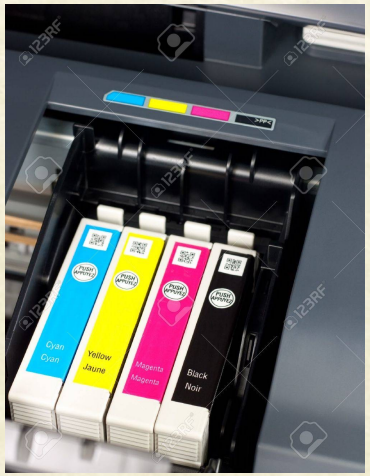


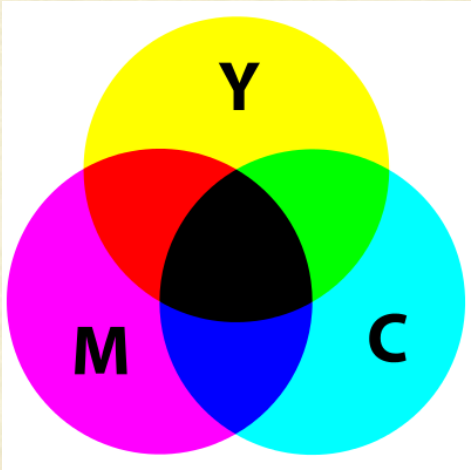
Magenta = Red + Blue
Cyan = Blue + Green
Yellow = Green + Red

Magenta = White - Green
Cyan = White - Red
Yellow = White - Blue

Why CMYK in Printing?

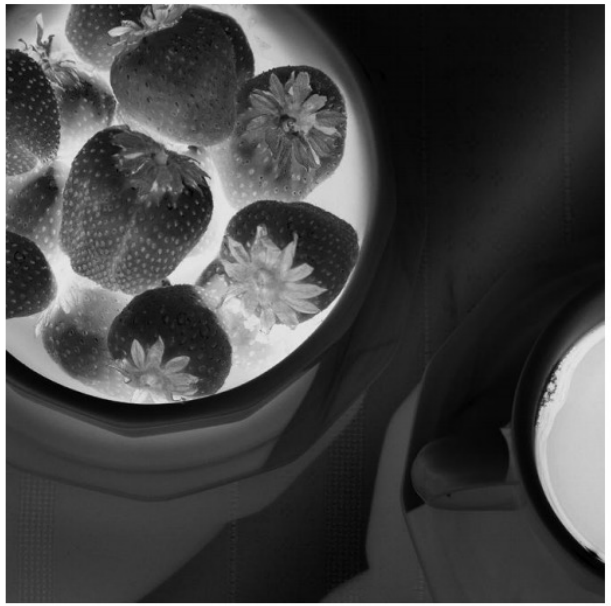
- Color pigments absorb specific wavelenths.
- CMY cover most lighter color ranges quite easily compared to RGB.
- BUT CMY by itself will not be able to create very deep dark colors (e.g. “true black,”)
- Black (designated “K” for “key color”) is added to CMY to achieve a much wider range of colors.



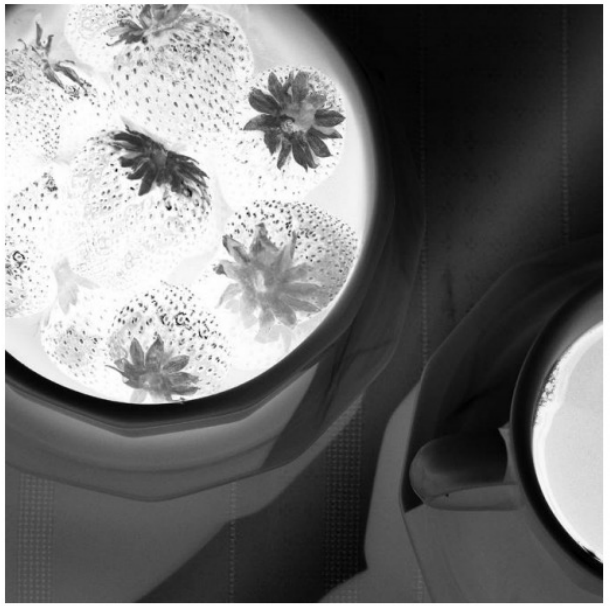


CMYK

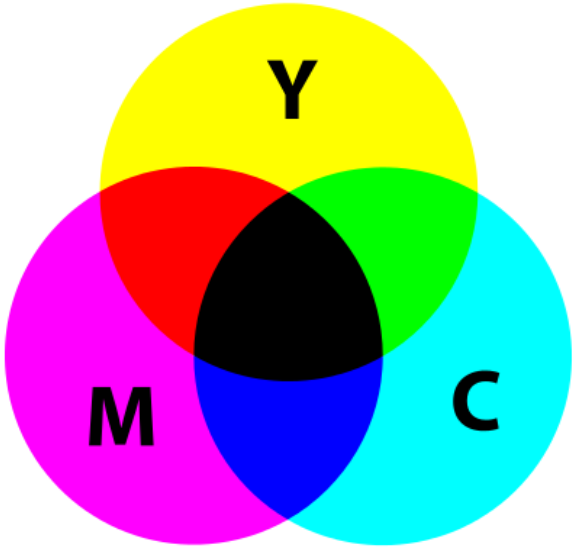
Full color

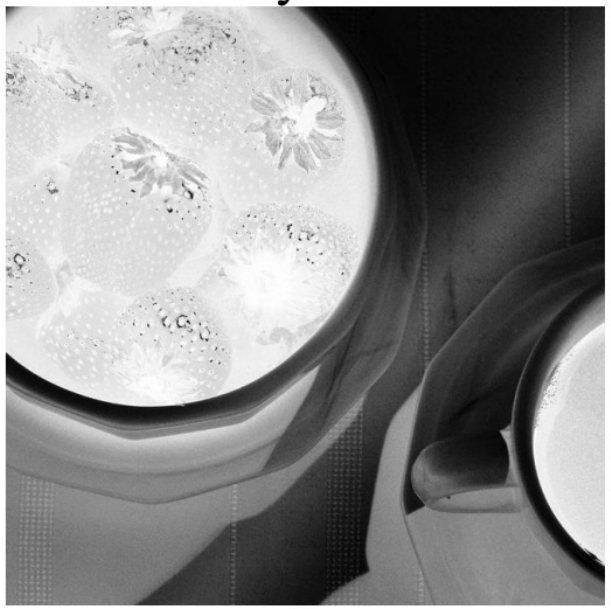


Cyan

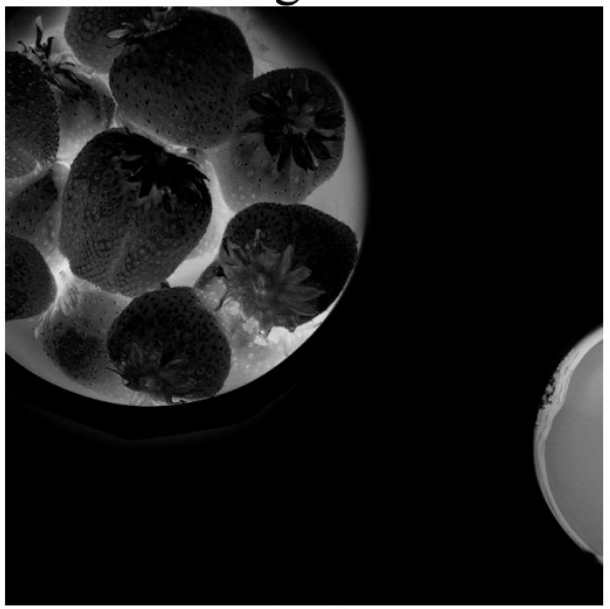


Magenta



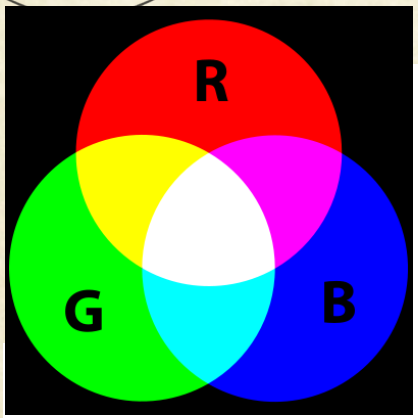


Yellow

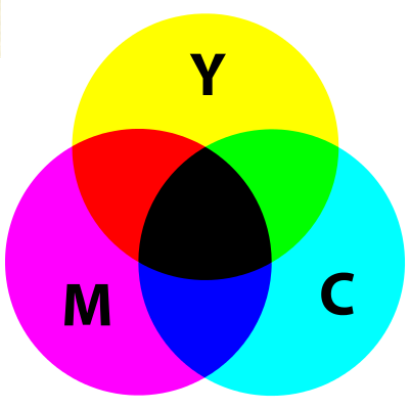


Black

RGB to CMY



$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

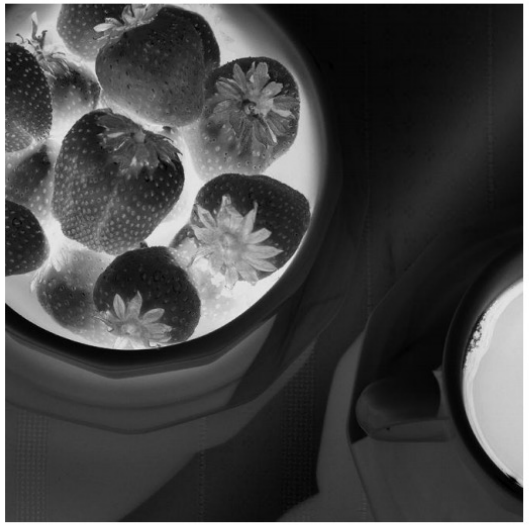


Full color

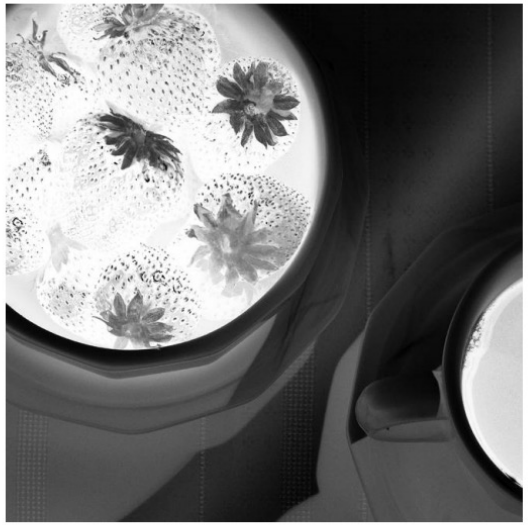
Red

Green

Blue

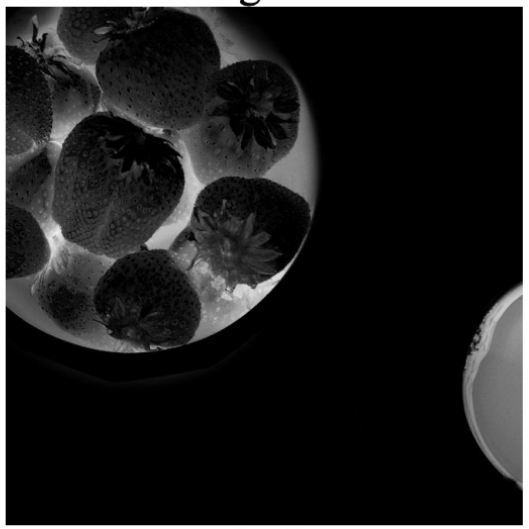


Cyan



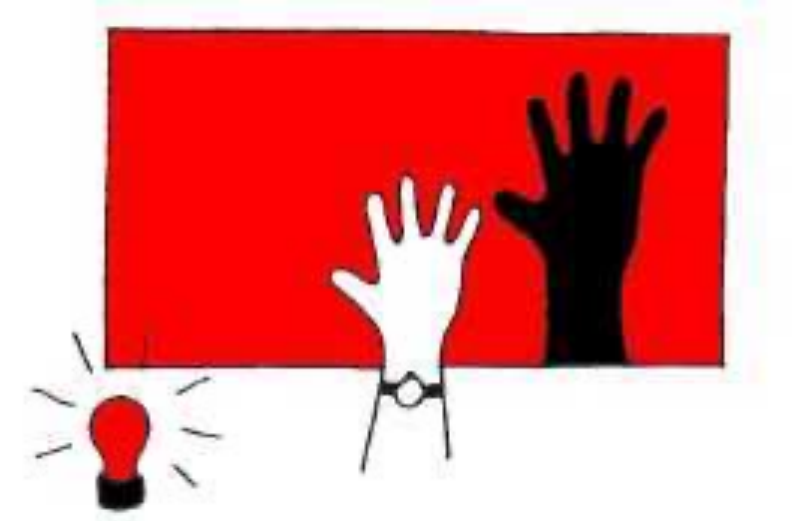
Magenta

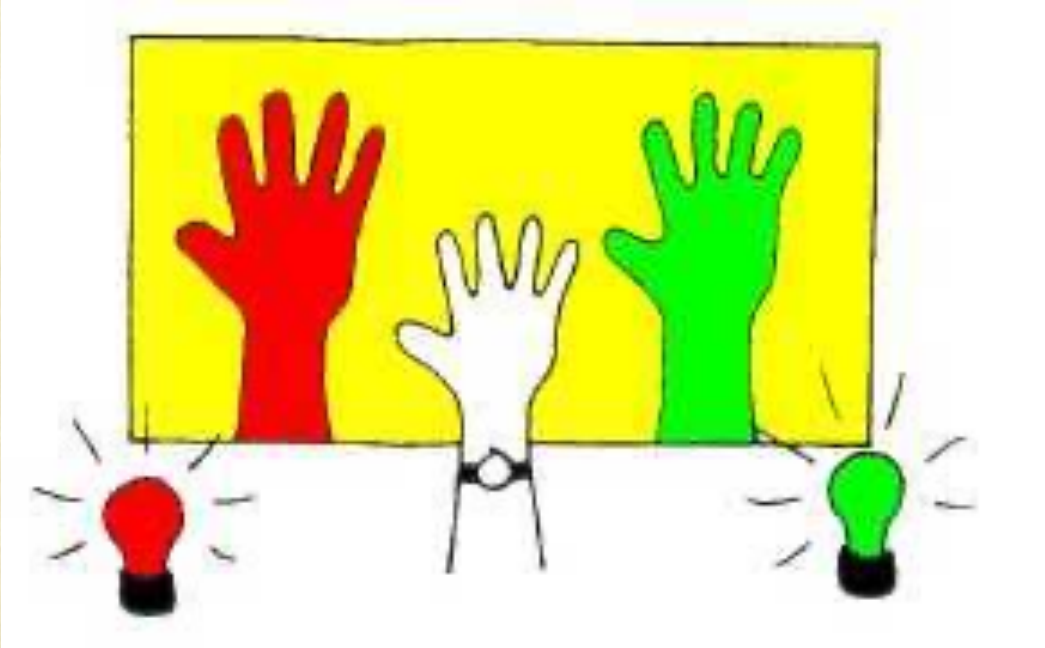
Yellow



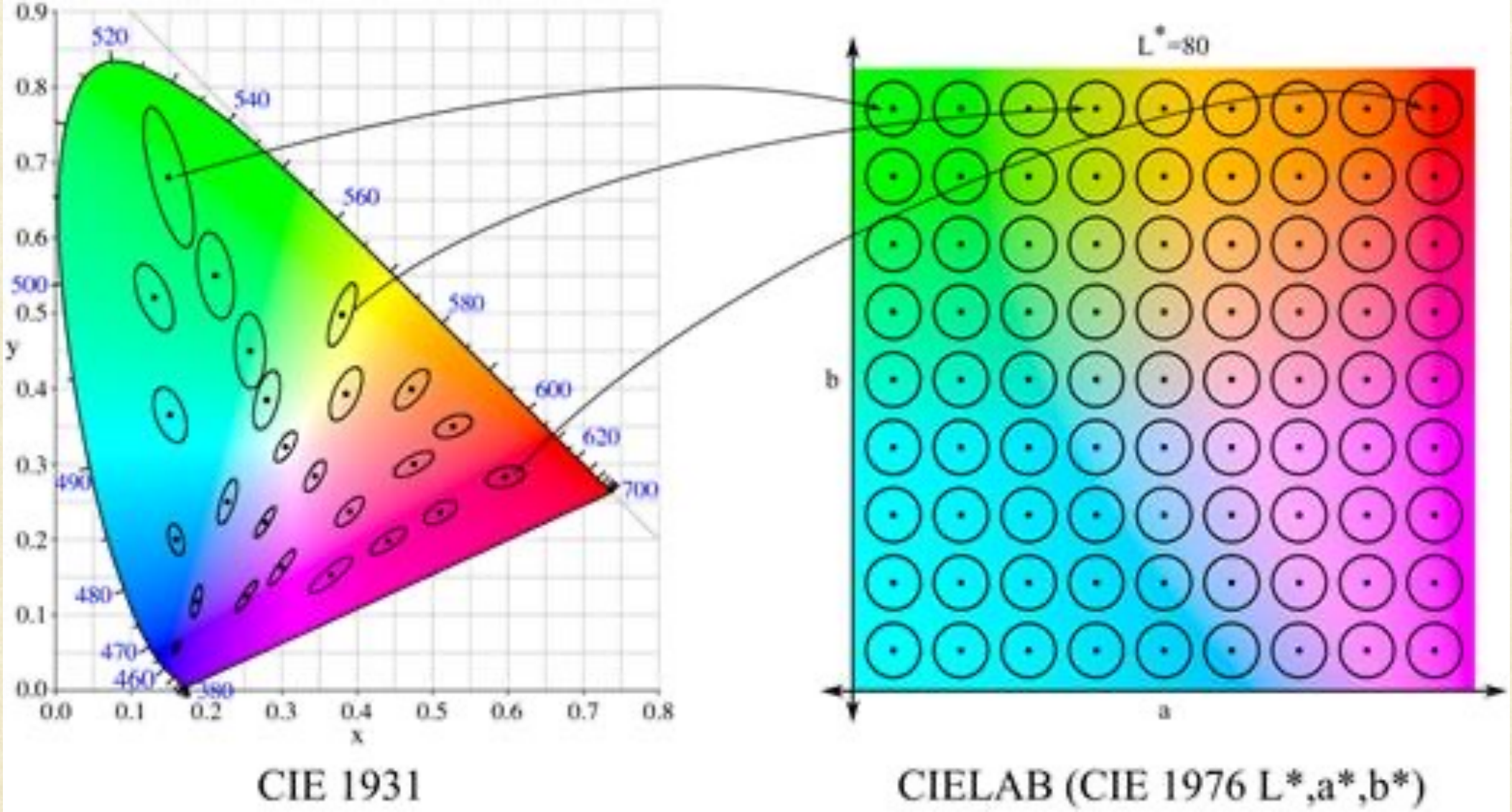
Black

RGB vs. CMY





CIE Lab color space



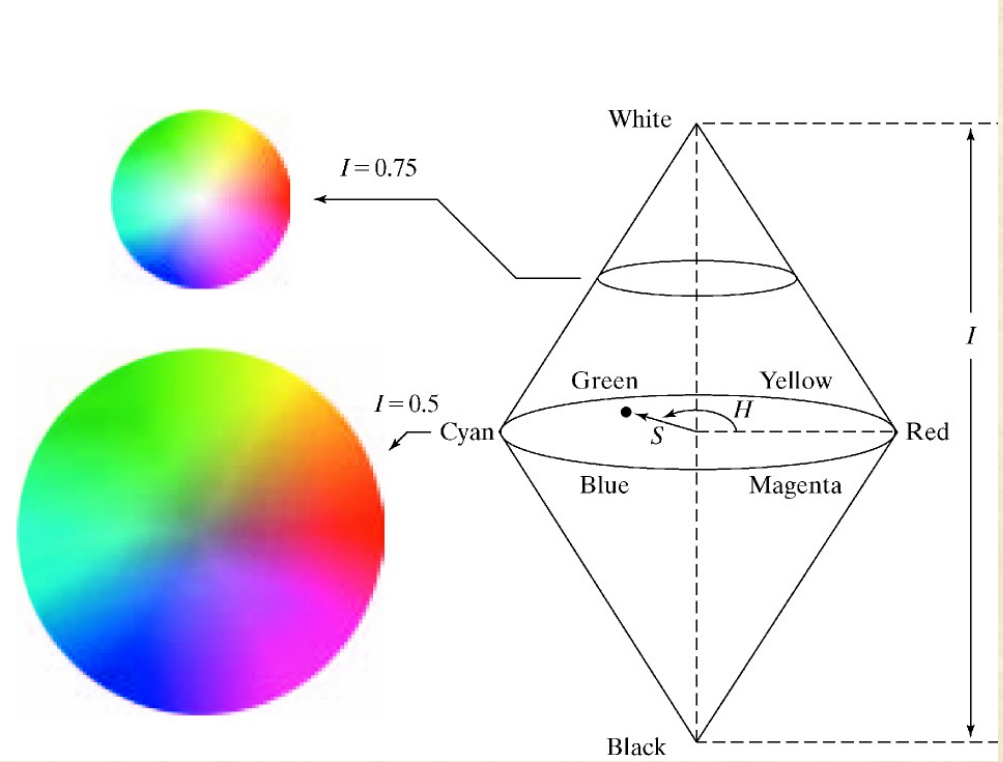
Ideal scenario

HSI color space

- Hue (H):
  - Dominant colour perceived by an observer
  - When we call an object red or orange, we refer to its hue.

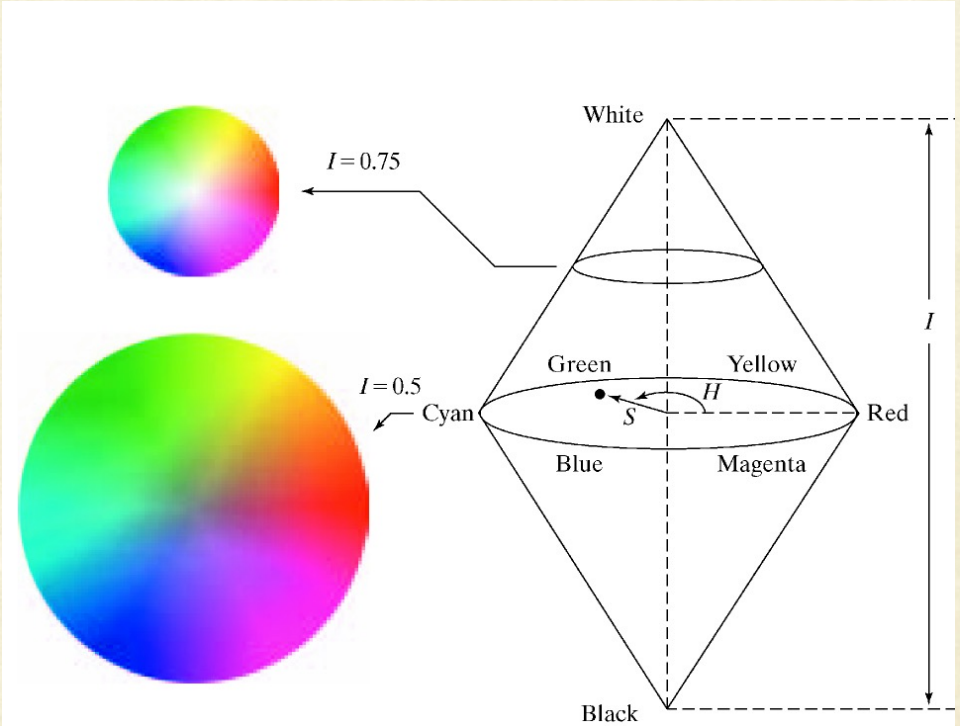
- Saturation (S):
  - Amount of white light mixed with a hue
  - Pure colors are fully saturated.
    - Pink (red+white) is less saturated.

- Brightness (I) : achromatic notion of intensity



HSI color space

- **HUE**
  - A subjective measure of color
  - Average human eye can perceive ~200 different colors
- **Saturation**
  - Relative purity of the color. Mixing more "white" with color reduces its saturation.
  - Pink has the same **hue** as **red** but less **saturation**
- **Intensity**
  - The brightness or darkness of an object



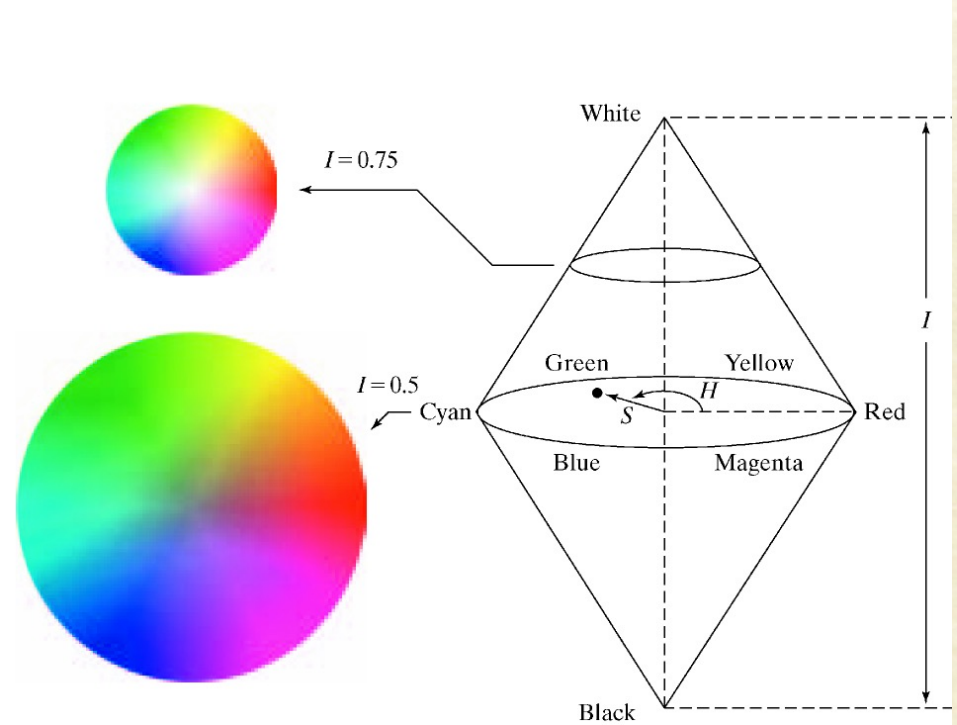
RGB : great for color generation
HSI : great for color description

HSI color space

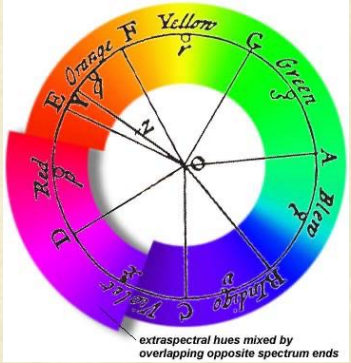
- **Hue** is defined as an angle
  - 0 degrees is **RED**
  - 120 degrees is **GREEN**
  - 240 degrees is **BLUE**

Saturation = % of distance from center : [0,1]

- **Intensity** is denoted as the distance "up" the axis from black.
  - Values range from 0 to 1



RGB : great for color generation
HSI : great for color description

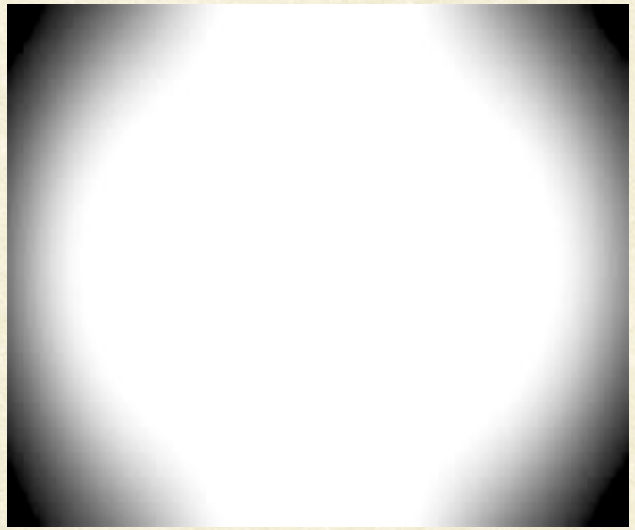


Example: Vintage effect or Sepia Toning

outR= (R * .293) + (G *.769) + (B * .210);
outG = (R * .249) + (G *.686) + (B * .188);
outB = (R * .172) + (G *.534) + (B * .151);

Example: Vignetting effect

×

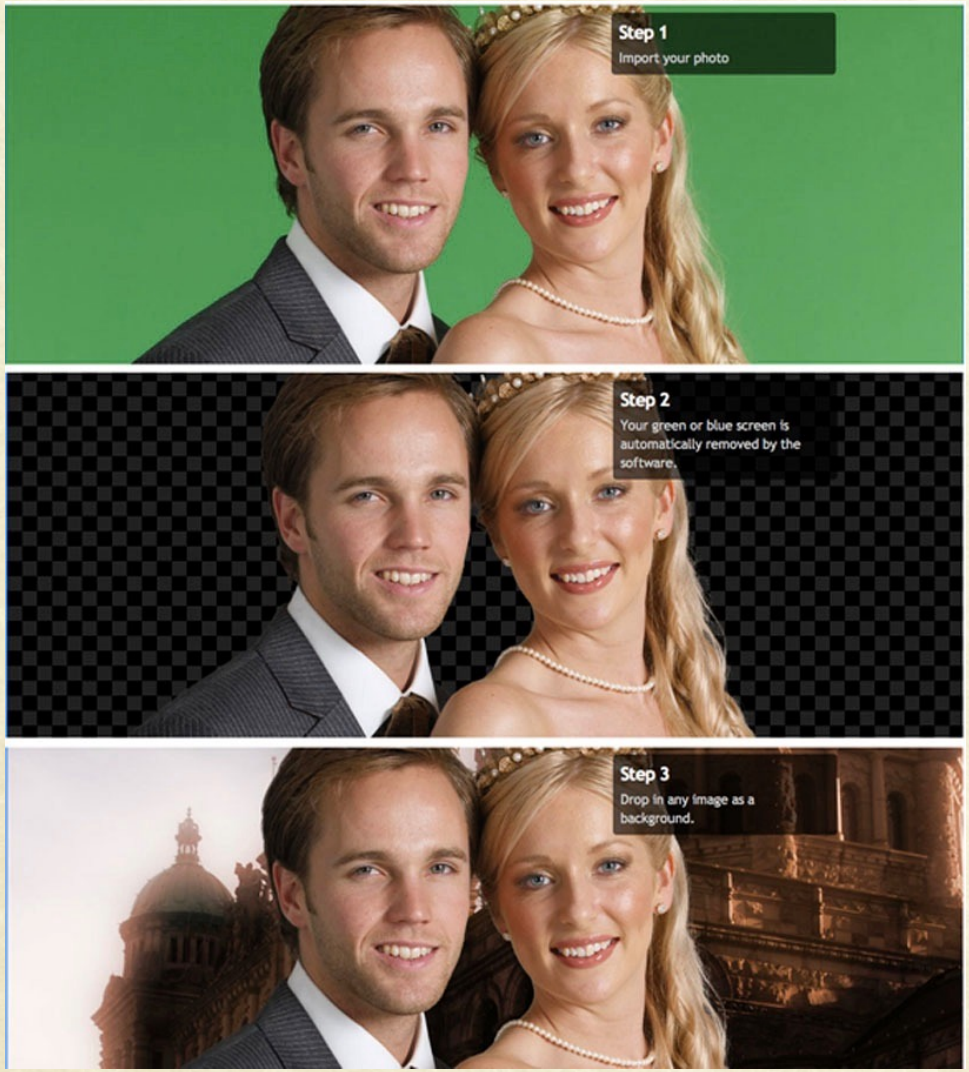


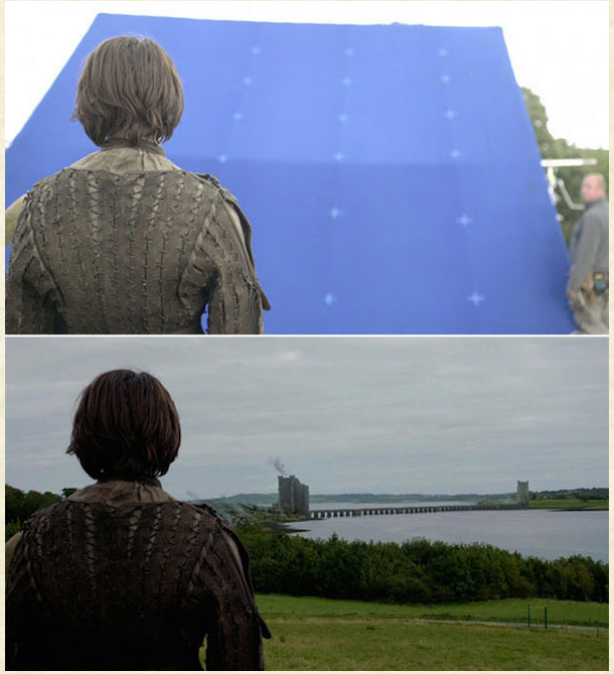
=

```
txt = imread(texture_path);
txt = imresize(txt,[size(out,1) size(out,2)]);
txt = double(rgb2gray(txt))/255;
```

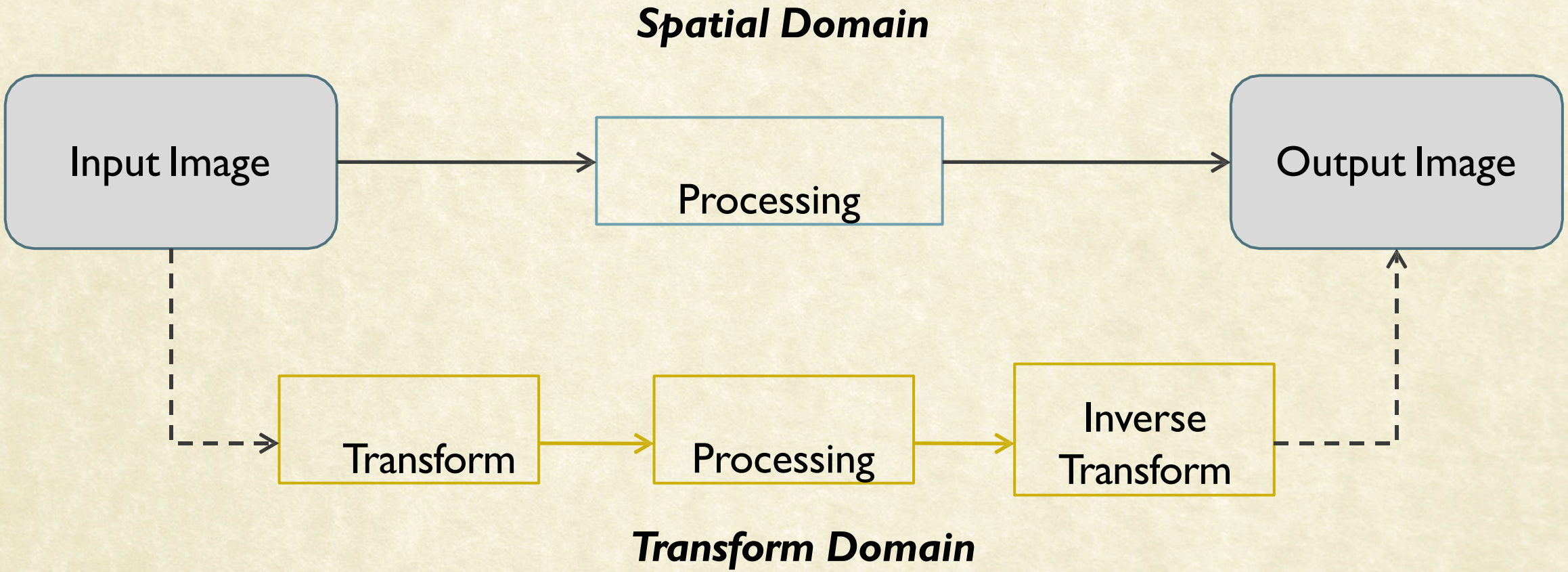
```
out1(:,:,1) = double(out(:,:,1)) .* double(txt);
out1(:,:,2) = double(out(:,:,2)) .* double(txt);
out1(:,:,3) = double(out(:,:,3)) .* double(txt);
```

Chroma Keying





Spatial vs. Transform Domain Processing



Spatial vs. Transform Domain Processing

Bandhani / Bandhej

Tie & Dye

Spatial vs. Transform Domain Processing

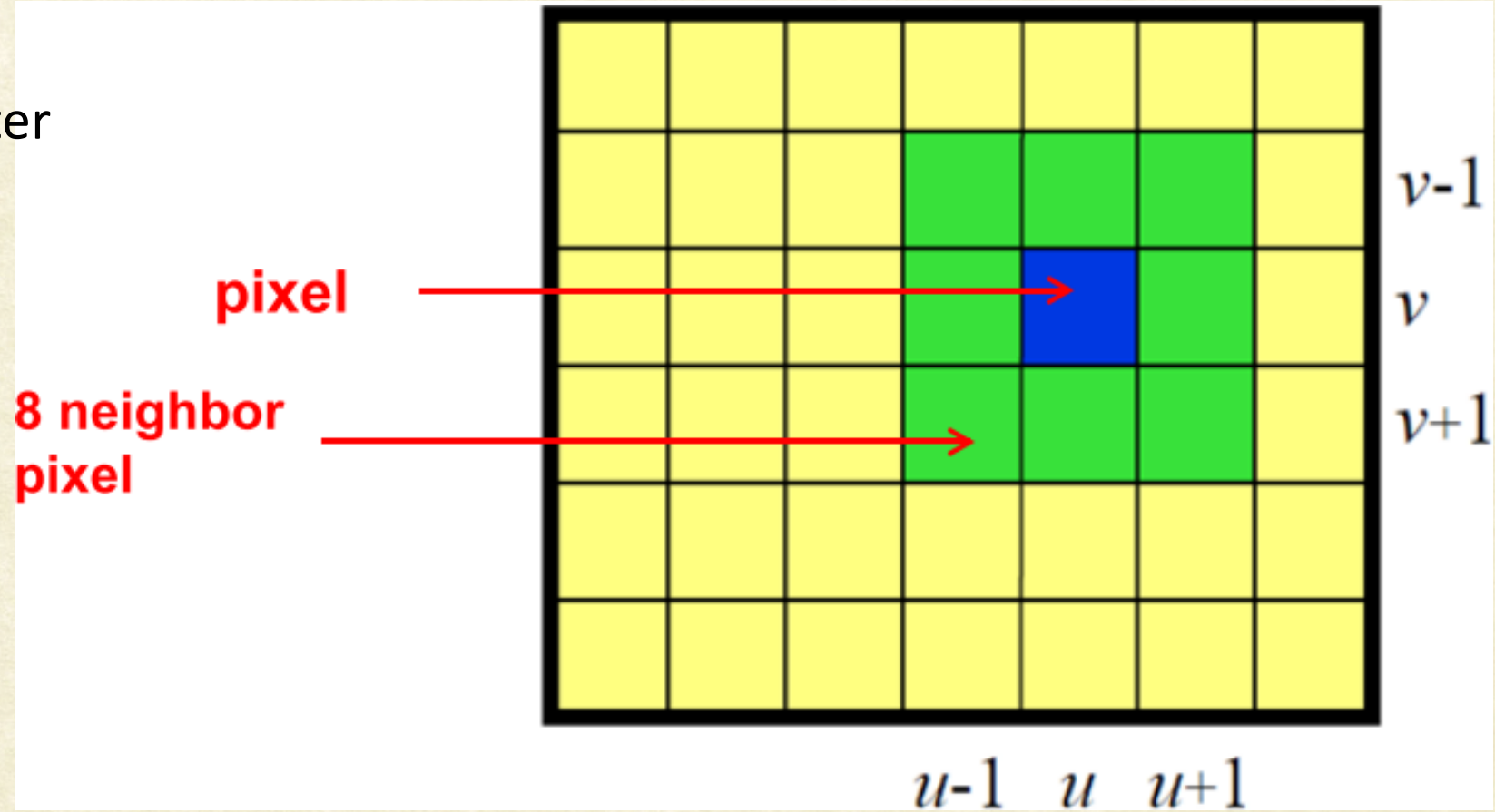
Transform (Tie)

Process (Dye)

Inverse Transform (Untie)

Spatial Domain Filtering

2 components of a spatial filter
- neighborhood
- a predefined operation



Filter aka
- Kernel
- Spatial Mask
- Template
- Window

Mean/Average Filter (Smoothing)

```
M = 3
For each valid location [x,y] in S
        a ← Average of intensities in a M x M neighborhood centered on [x,y]
        D[x,y] = round(a)
```

120	190	140	150	200
17	21	30	8	27
89	123	150	73	56
10	178	140	150	18
190	14	76	69	87

x

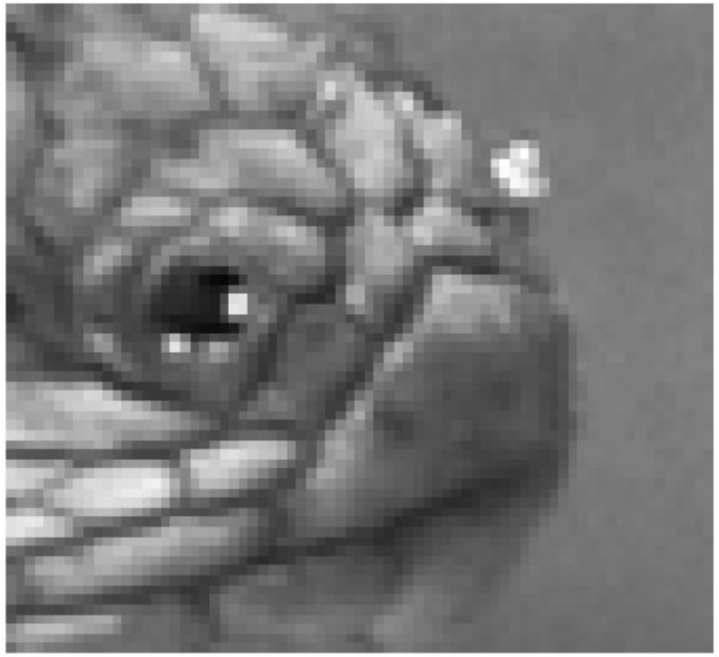
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

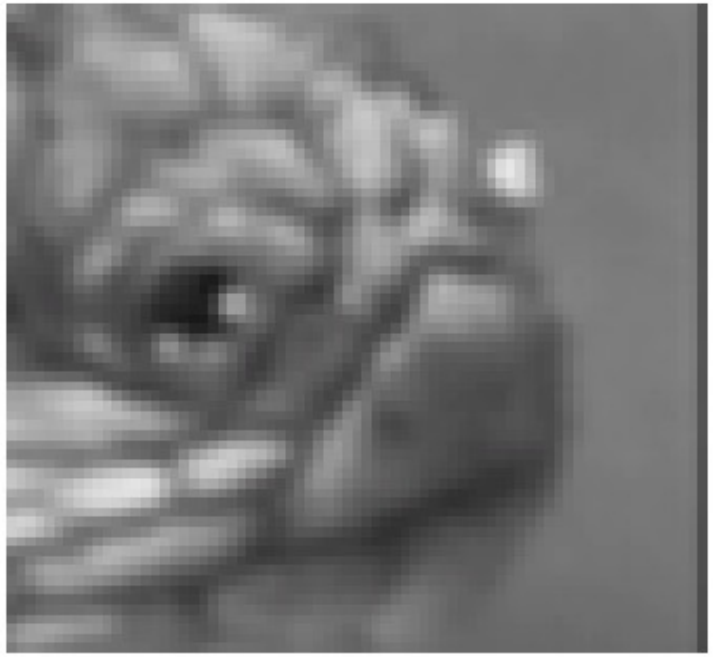
=

	98			

http://199.116.233.101/index.php?title=File:Animated-kernels.gif

$$1/9 * \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

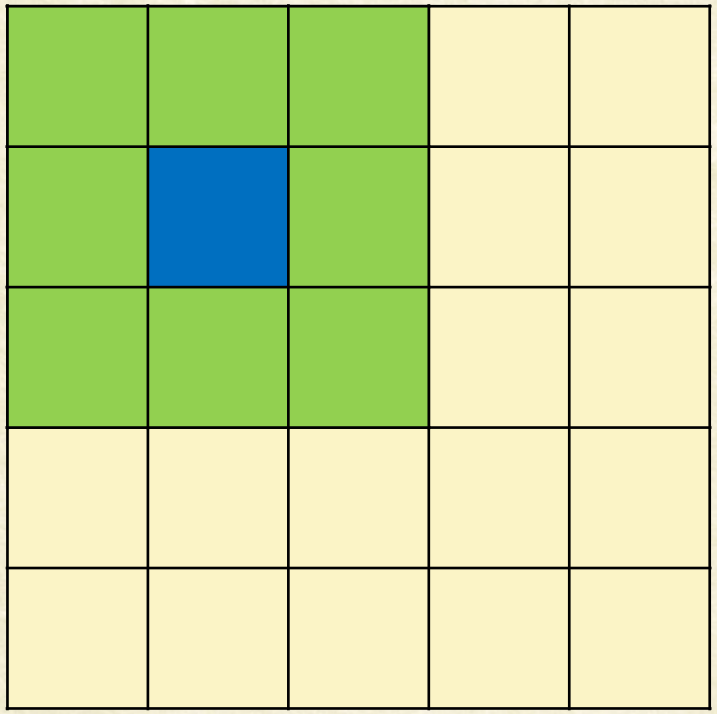




Mean/Average Filter

Note: Coefficients sum to 1

I



H

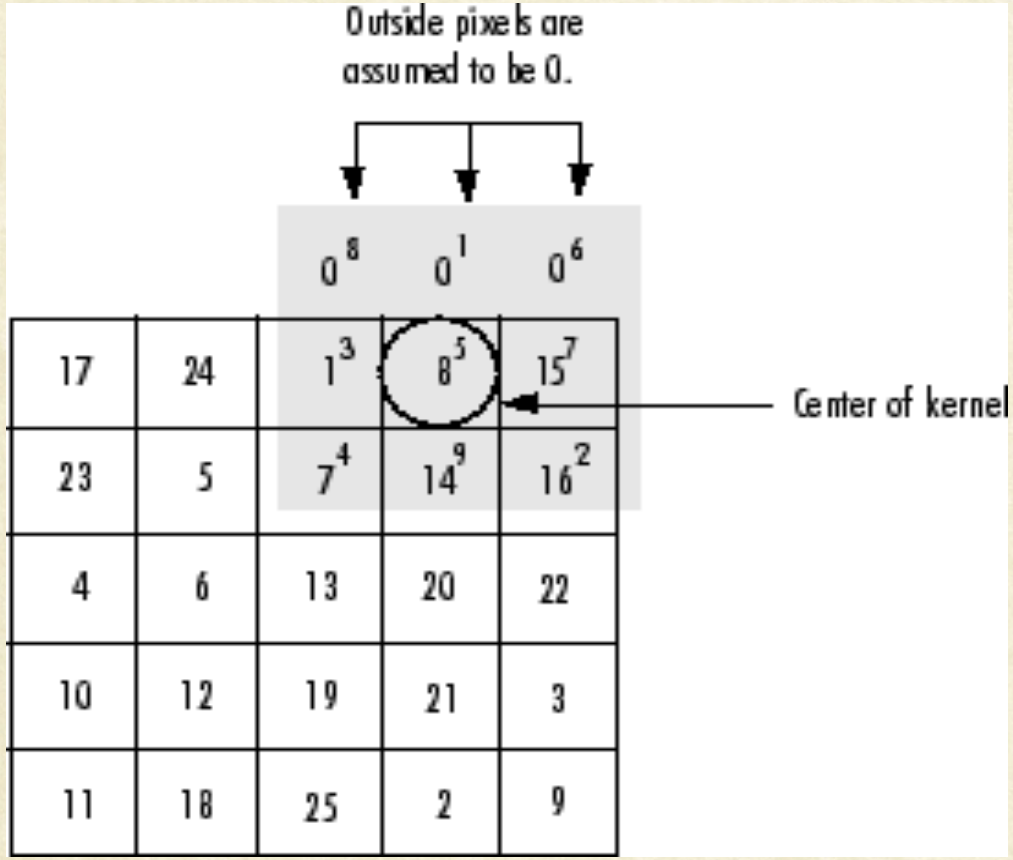
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3 x 3 Weight Mask / Kernel / Filter

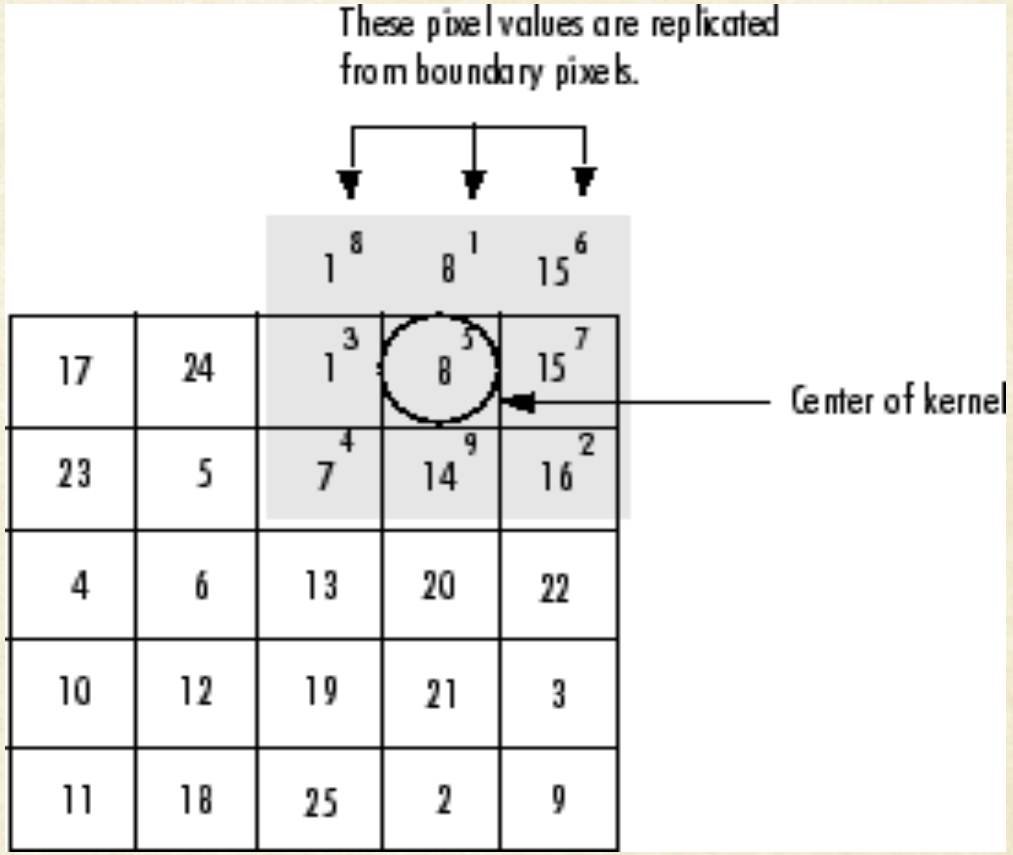
$$I'(u,v) \leftarrow \frac{1}{9} \cdot \sum_{j=-1}^{1} \sum_{i=-1}^{1} I(u+i, v+j)$$

$$I'(u,v) \leftarrow \sum_{j=-1}^{1} \sum_{i=-1}^{1} I(u+i, v+j) \cdot \boxed{H(i,j)}$$

Image Padding



Zero/Constant



Replicate/Reflect

Effect of Mask Size

Original Image

[3x3]

[5x5]

[7x7]

Weighted Averaging

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Standard average

Weighted Average

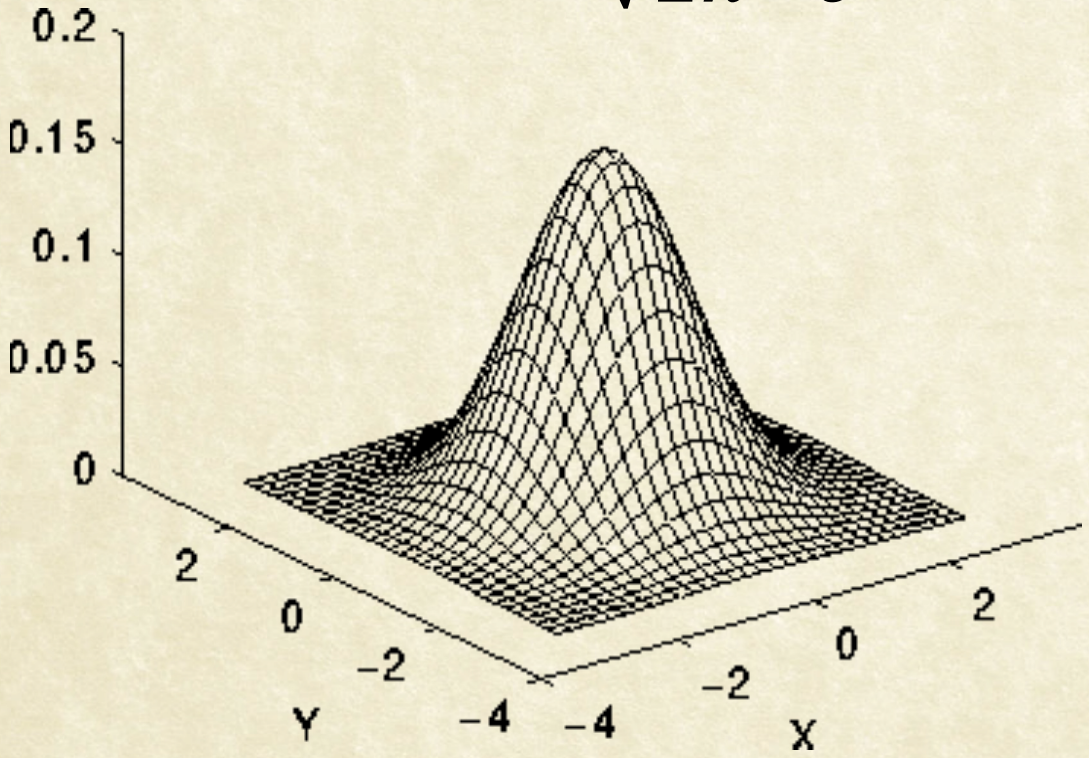
$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$I'(u,v) \leftarrow \frac{\sum_{j=-a}^{a} \sum_{i=-b}^{b} I(u+i, v+j) \cdot H(i,j)}{\sum_{j=-a}^{a} \sum_{i=-b}^{b} H(i,j)}$$

Gaussian Smoothing

- Mask weights are samples of a zero-mean 2-D Gaussian

$$G(x,y) = \frac{1}{\sqrt{2\pi}\cdot\sigma} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



$\frac{1}{265}$

1	4	6	4	1
4	16	26	16	4
6	26	43	26	6
4	16	26	16	4
1	4	6	4	1

5×5 Gaussian filter, $\sigma = 1$

Image Gradient and Edges

$$\frac{f(x+h,y)-f(x-h,y)}{2h} \Rightarrow \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

x-derivative

$$\frac{f(x,y+h)-f(x,y-h)}{2h} \Rightarrow \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

y-derivative

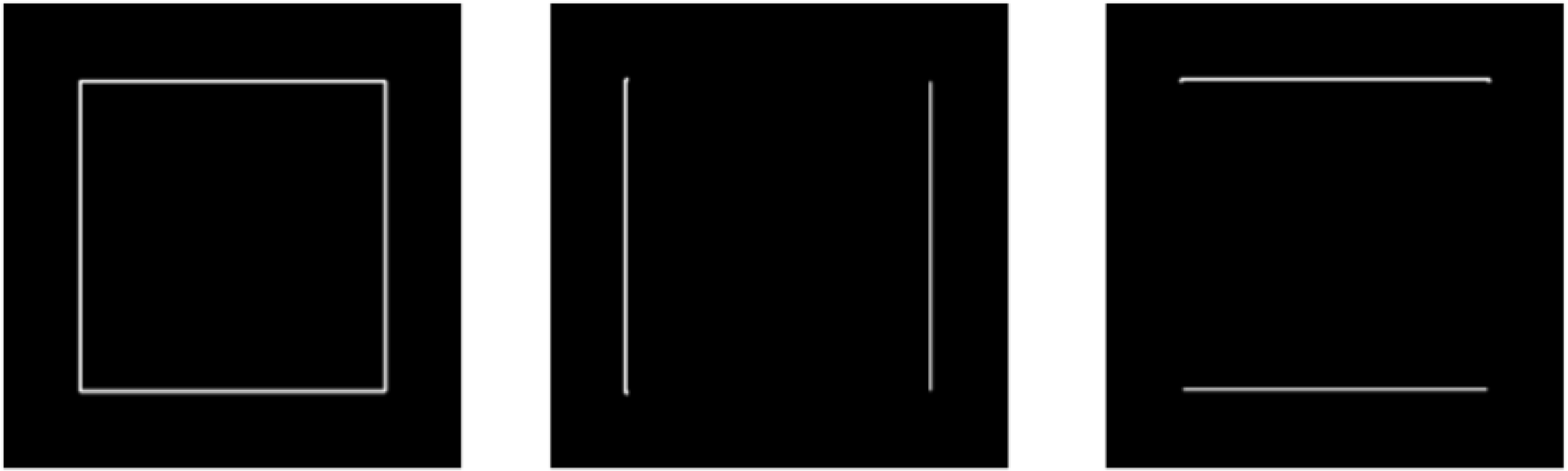


Image | Gradient in x | Gradient in y

https://i1.wp.com/theailearner.com/wp-content/uploads/2019/05/gradxy.png?resize=768%2C280&ssl=1

https://nihrecord.nih.gov/sites/recordNIH/files/pdf/1984/NIH-Record-1984-03-13.pdf

Prewitt Edge Filter

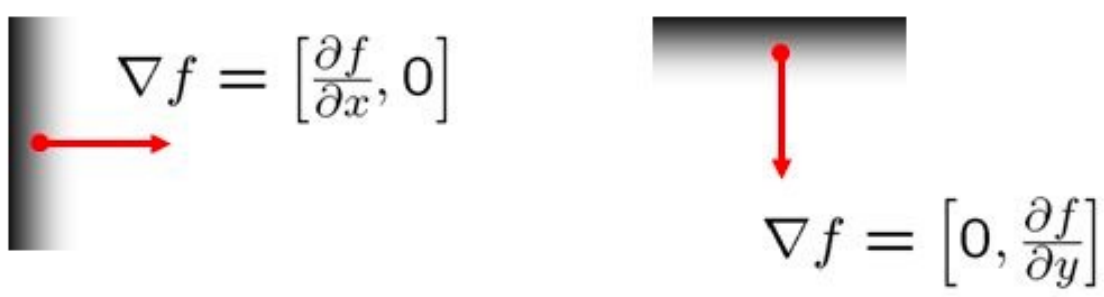
-1	0	+1
-1	0	+1
-1	0	+1

G_x

+1	+1	+1
0	0	0
-1	-1	-1

G_y

Edge is perpendicular to gradient

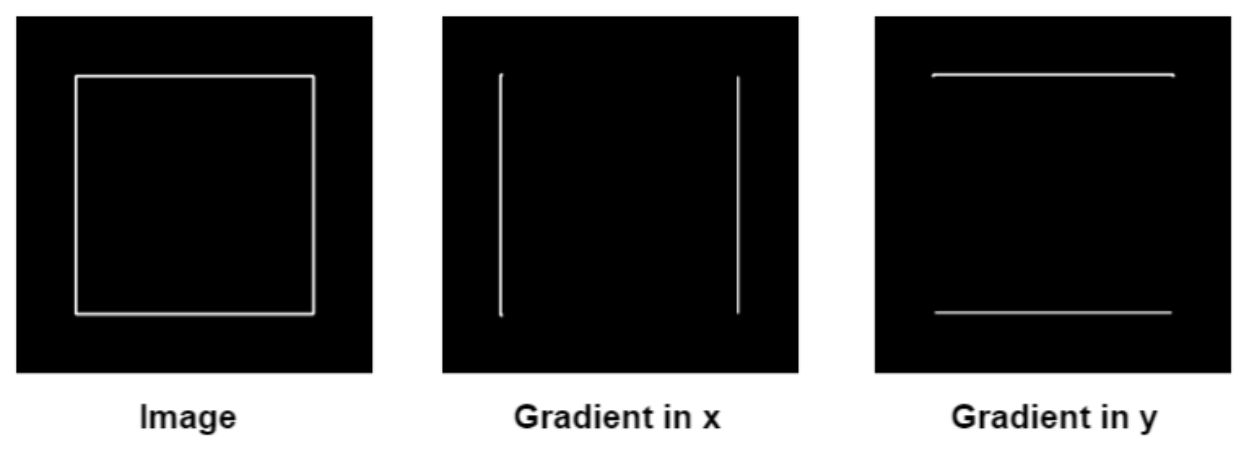


-1	0	+1
-1	0	+1
-1	0	+1

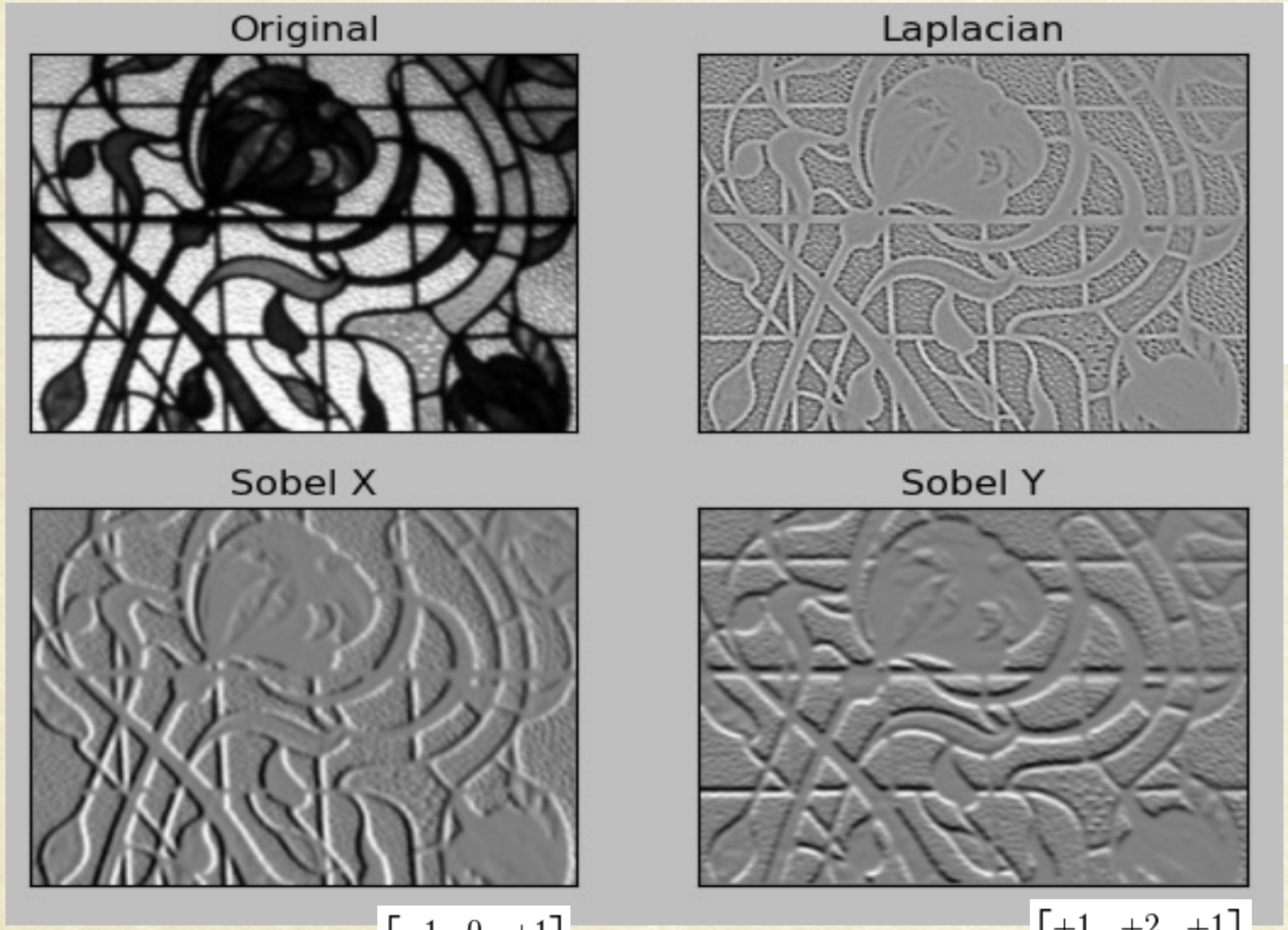
G_x

+1	+1	+1
0	0	0
-1	-1	-1

G_y



Edge Masks: Sobel, Laplacian



0	−1	0
−1	4	−1
0	−1	0

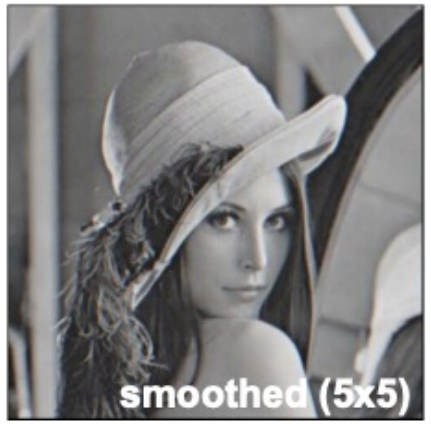
$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$\begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Highboost Filtering

- What does blurring take away?

–



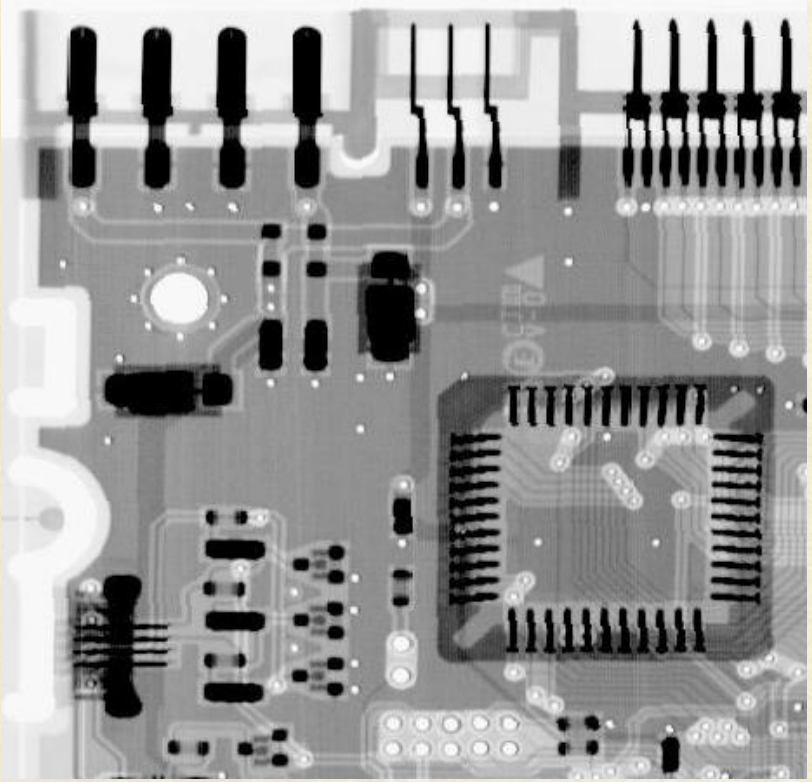
=

- Let's add it back:

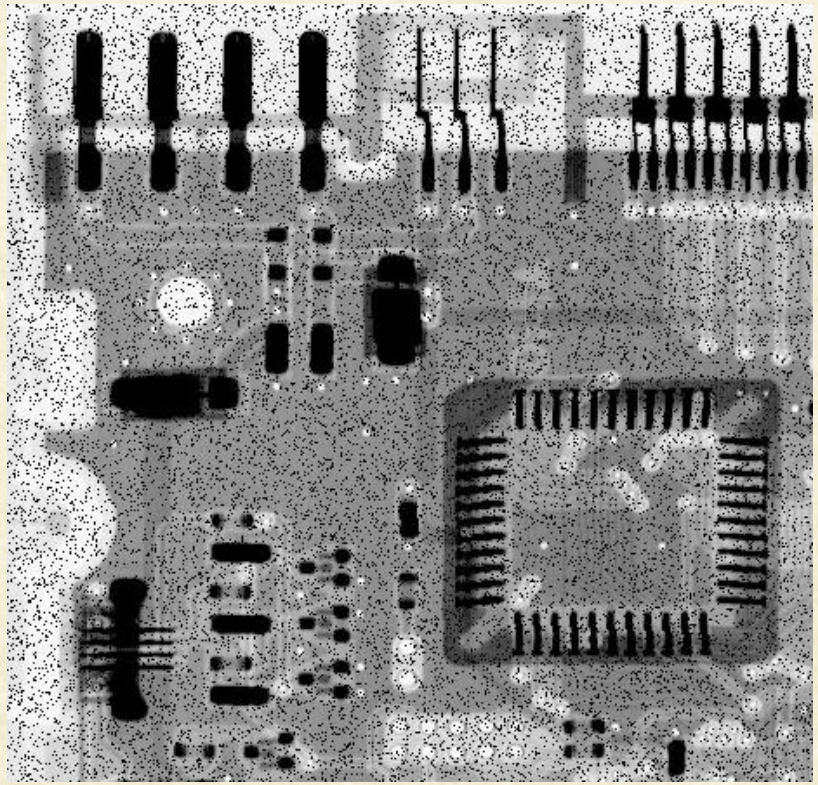
+ a

=

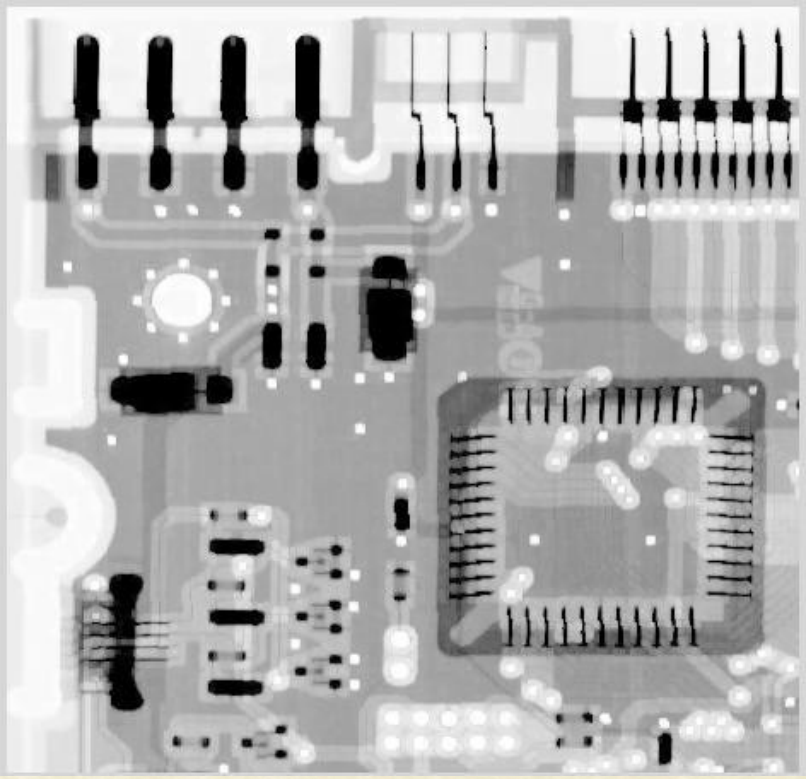
Non-linear Spatial Filters (max)



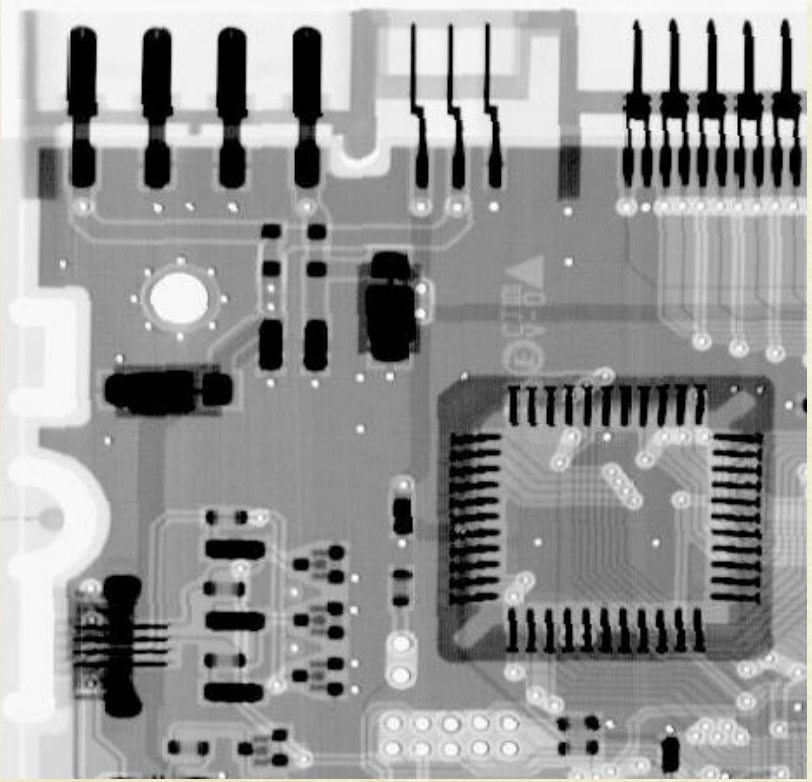
Pepper noise



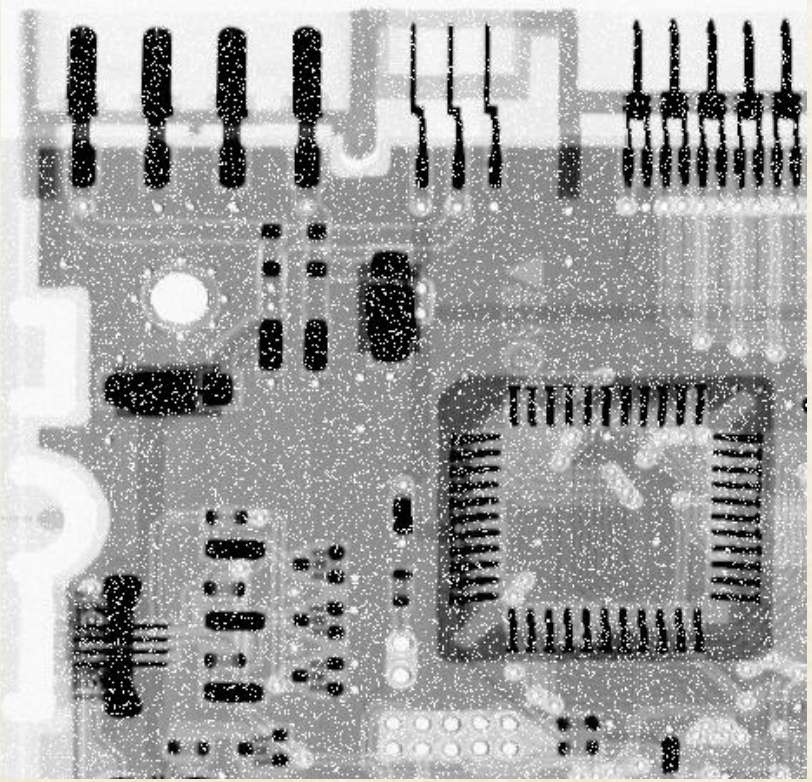
After applying max filter



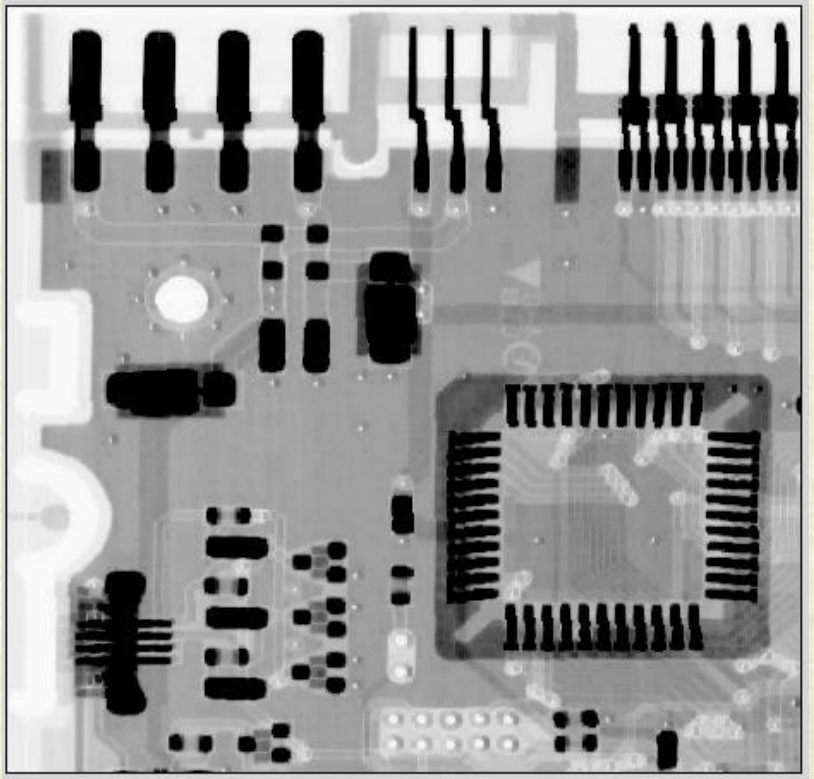
Non-linear Spatial Filters (min)



Salt noise



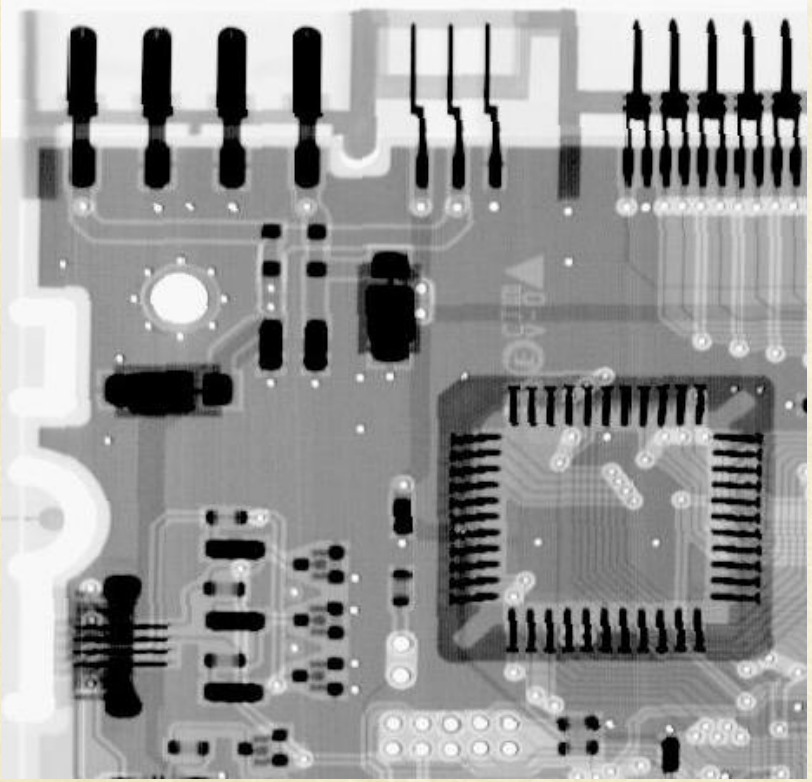
After applying min filter

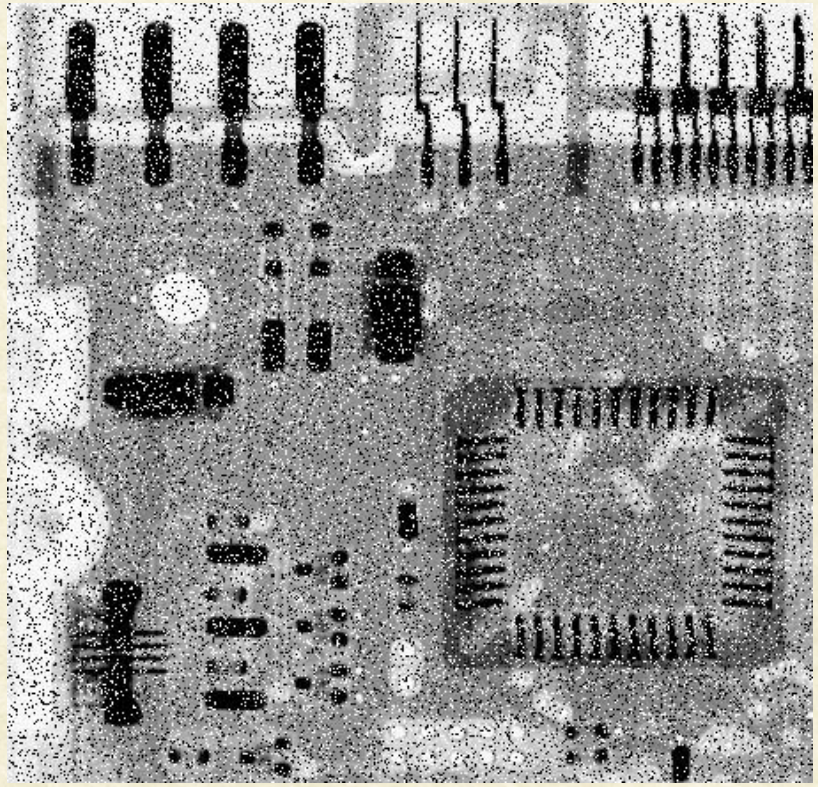


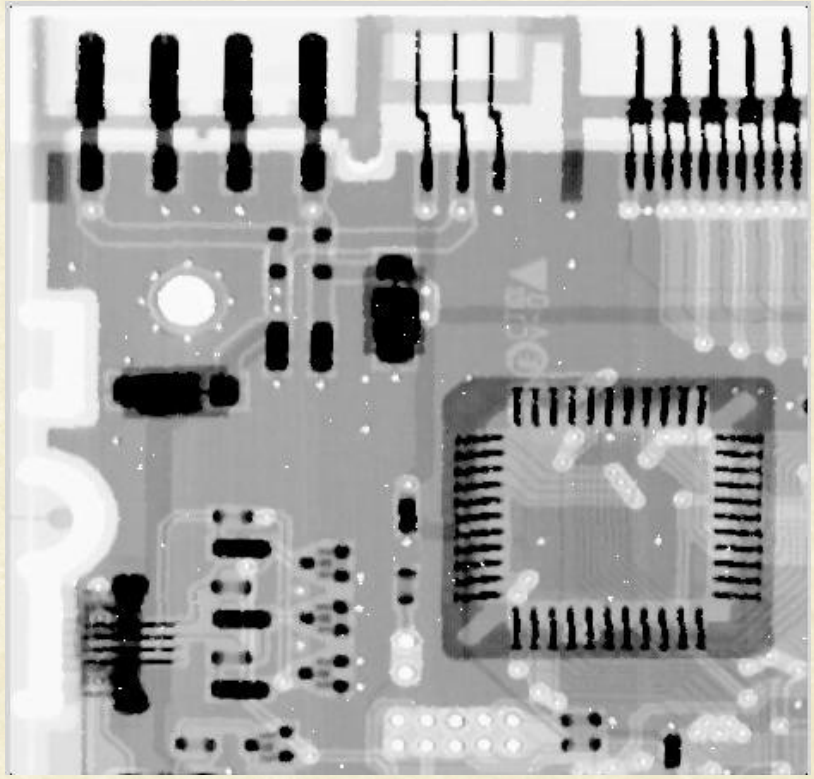
Non-linear Spatial Filters (median)

Salt & Pepper noise

After applying median filter







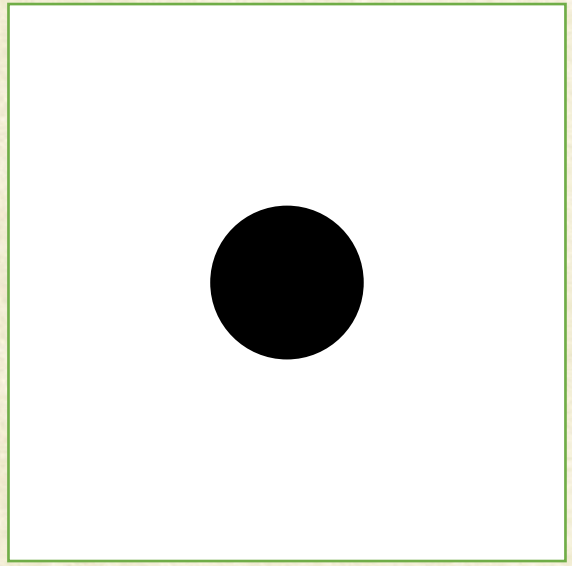
max, min, median → also known as rank / order statistic filters

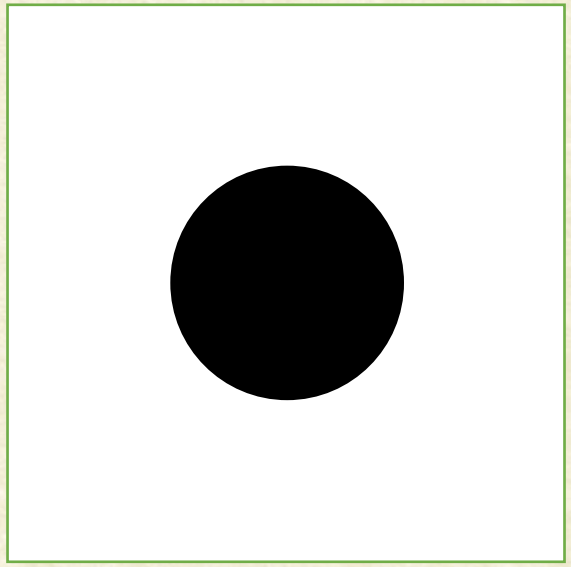
Other Spatial Filters

- Geometric mean
- Harmonic mean
- Contra harmonic mean
- Mid Point filter
- Alpha trimmed mean filter
- …..

CS7.404: Digital Image Processing

Monsoon 2023: Morphological Image Processing 1



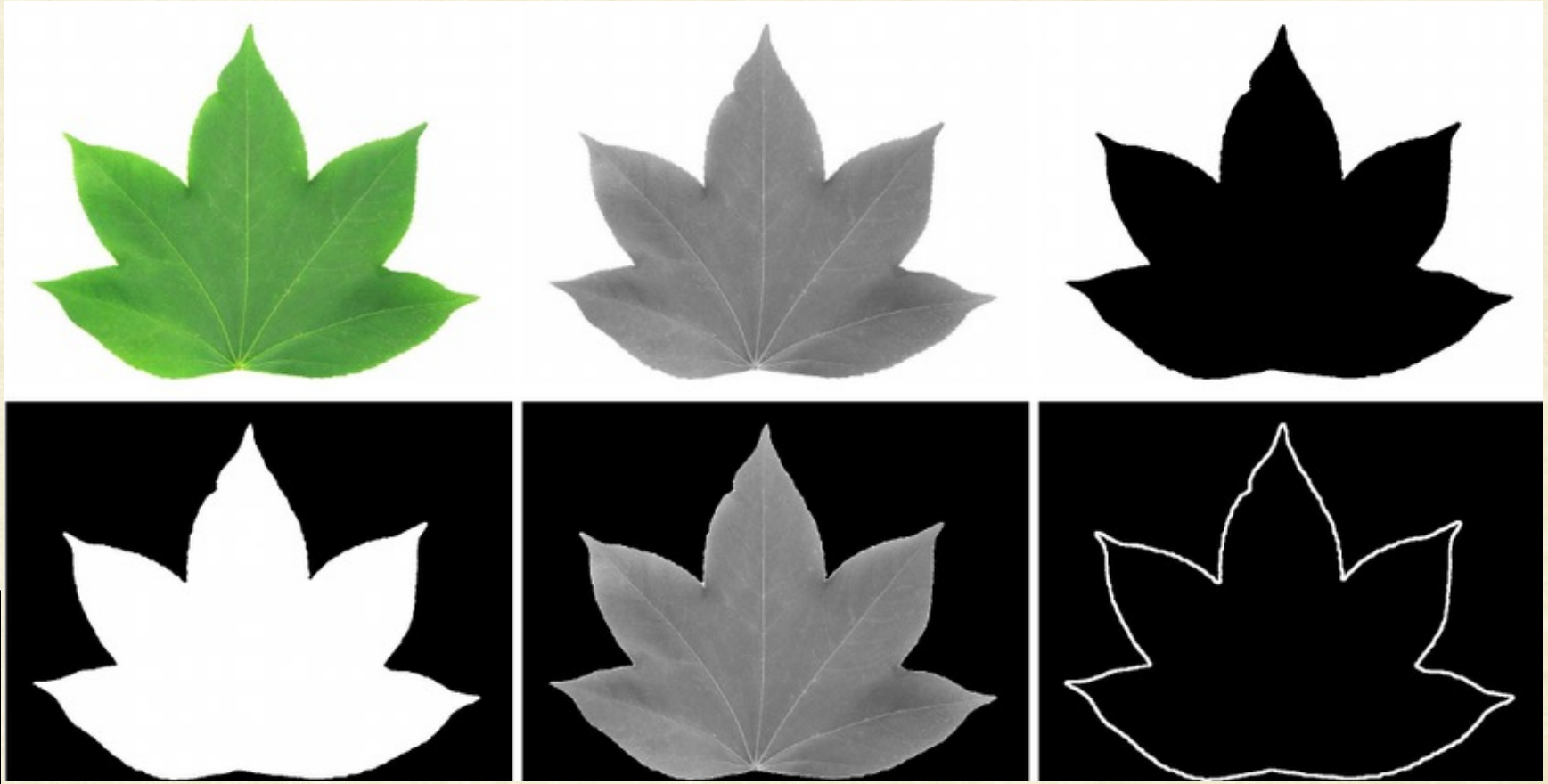


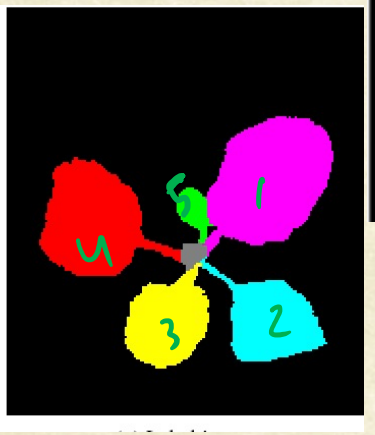
Anoop M. Namboodiri

Biometrics and Secure ID Lab, CVIT,

IIIT Hyderabad

Plant Phenotyping





Recognizing Scene Text

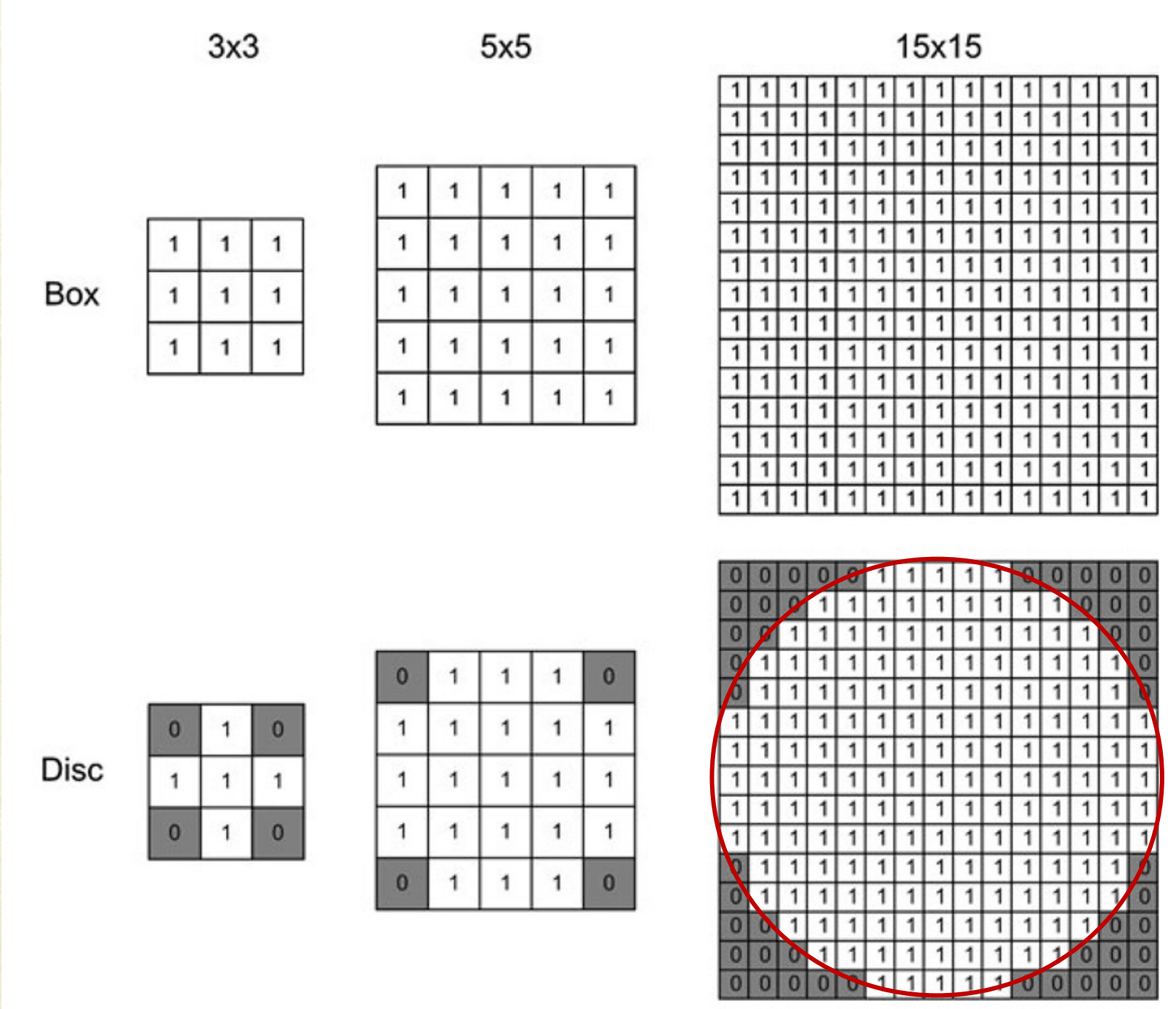
https://cvit.iiit.ac.in/research/projects/cvit-projects/scene-text-understanding

Image – Set of Pixels

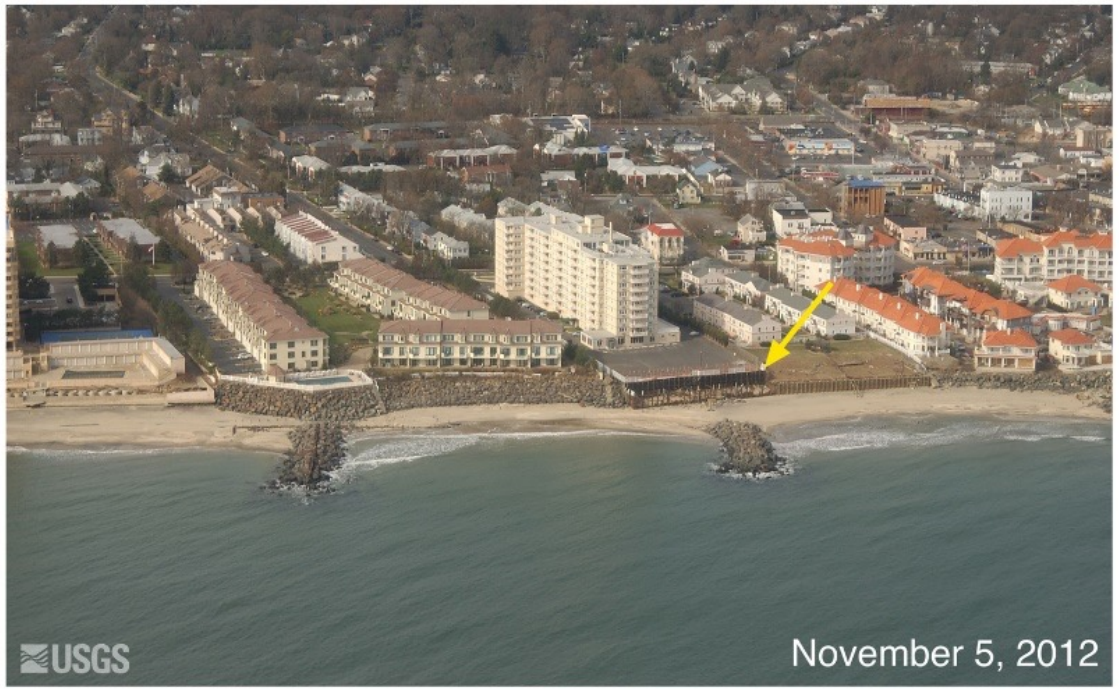
- Basic idea:
  - Object/Region = set of pixels (or coordinates of pixels)

- 0 = background
- 1 = foreground

Structuring Element



Erosion

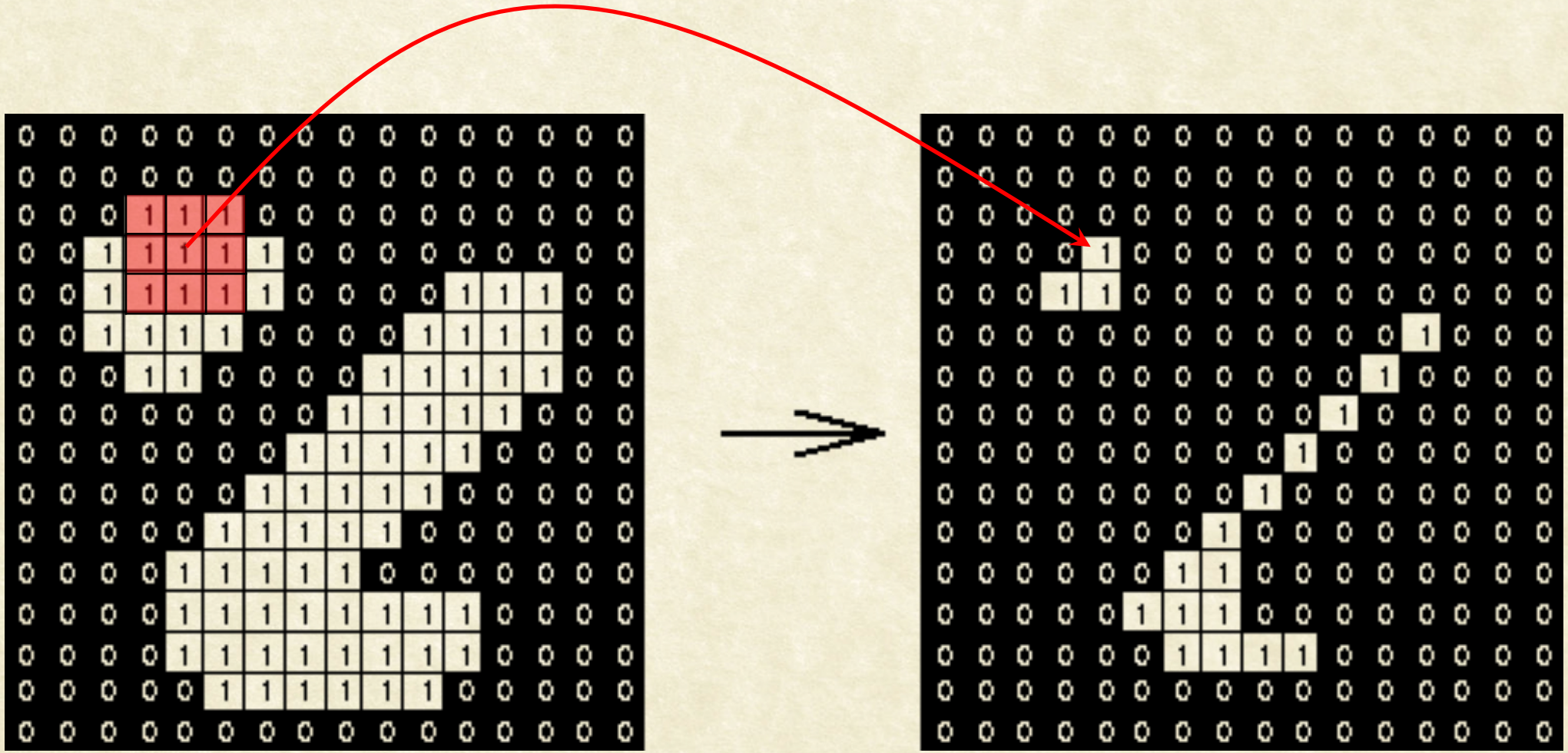


Erosion

Erosion: Effect

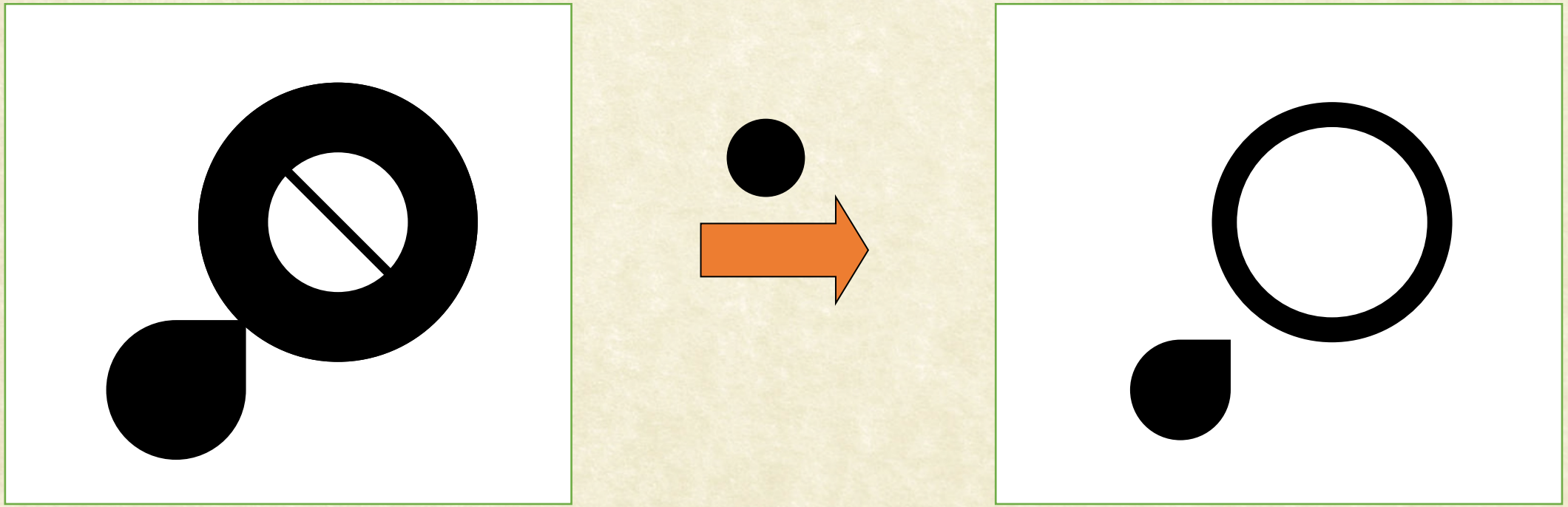
1	1	1
1	1	1
1	1	1

If, for a particular location of Structuring Element (SE) origin, SE lies **fully within the region**, retain the location, else set to 0



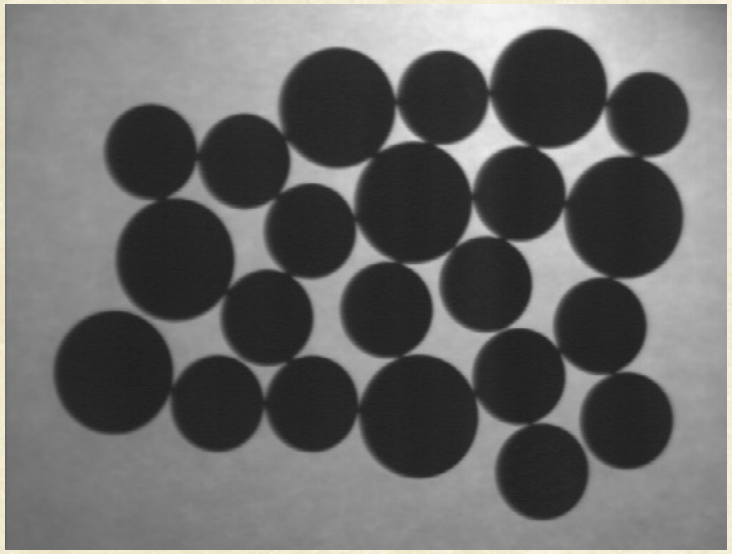
Erosion

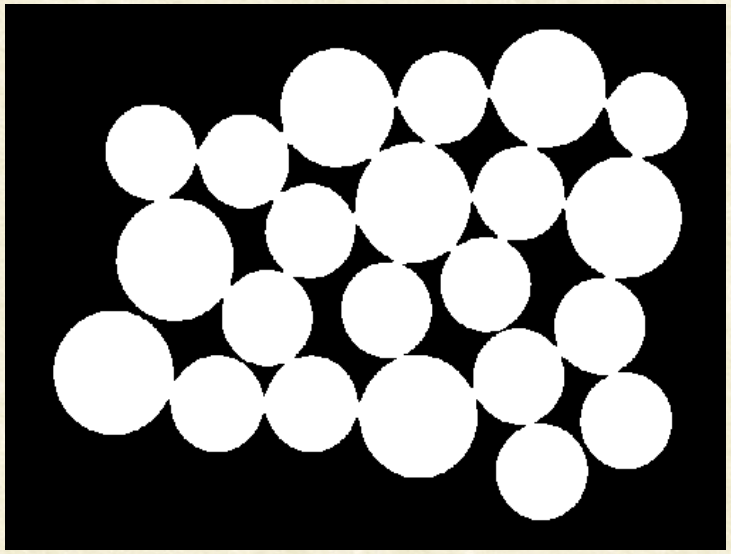
- Shrinks foreground objects
- Thin features are removed
- Touching objects are separated
- NOTE: Multiple iterations of dilation

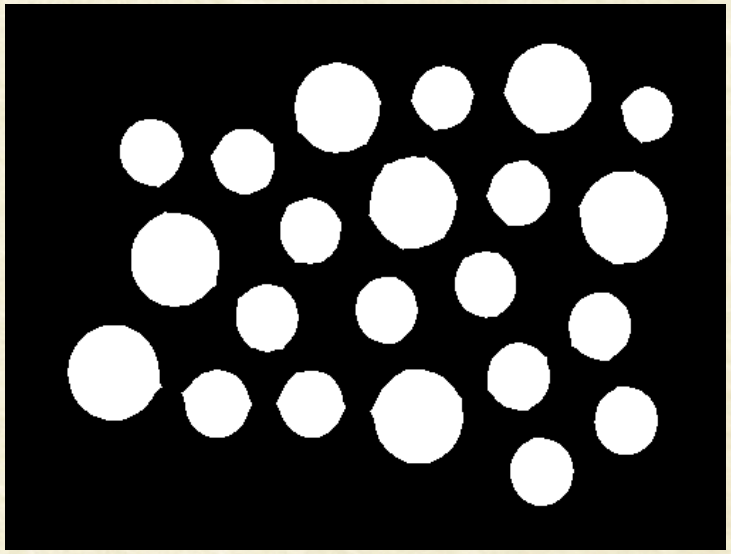


Example: Counting coins

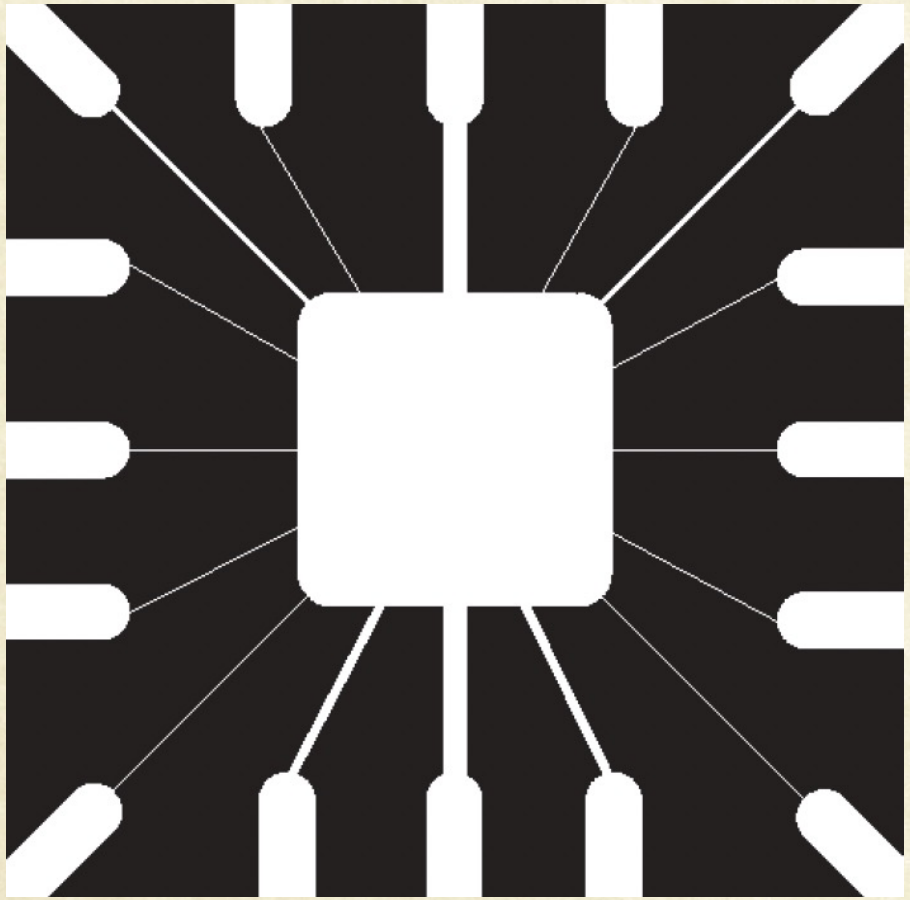
- Difficult because they touch each other!
- Solution: Binarization and Erosion separates them!



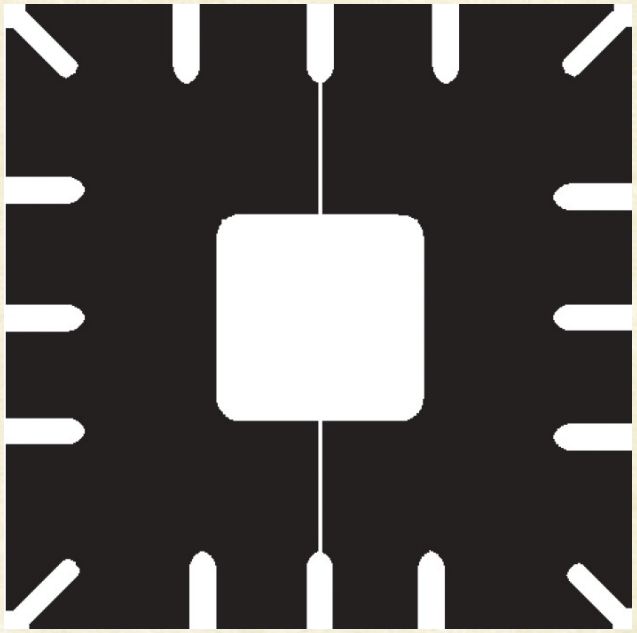




Erosion: Example



486 x 486 image

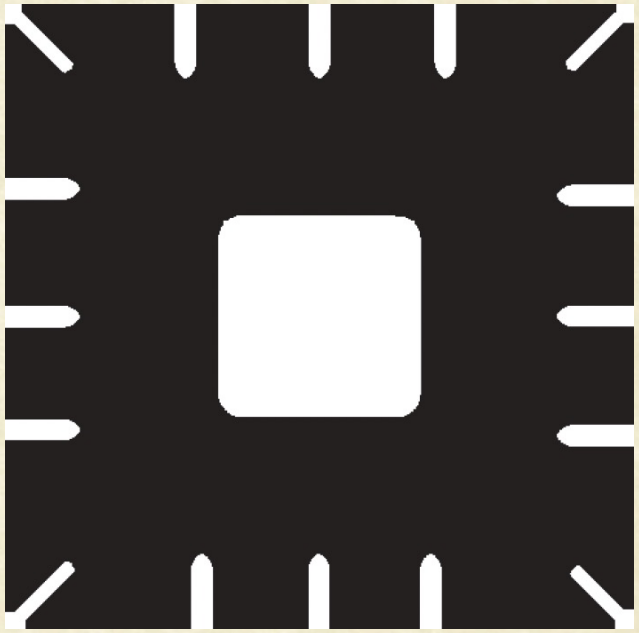


Erosion by square SE:

11 x 11

45 x 45

15 x 15



From: Digital Image Processing, Gonzalez,& Woods

Erosion operation and min filter

1	1	1
1	1	1
1	1	1

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

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

Dilation operation (max filter)

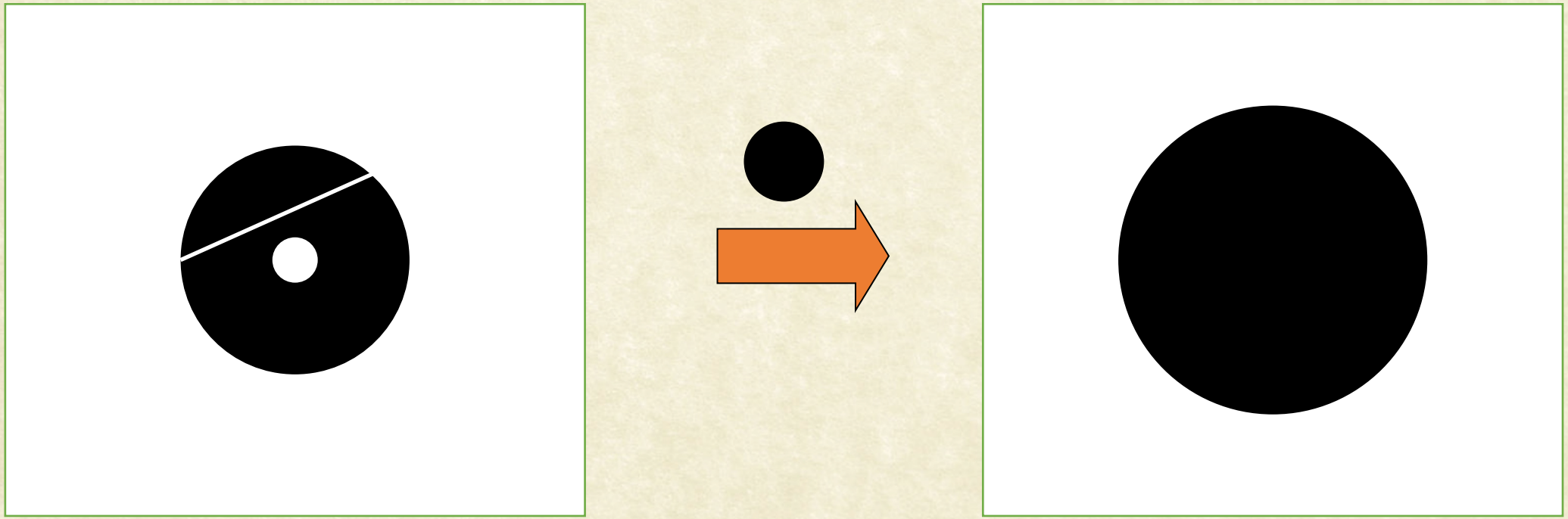
1	1	1
1	1	1
1	1	1

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

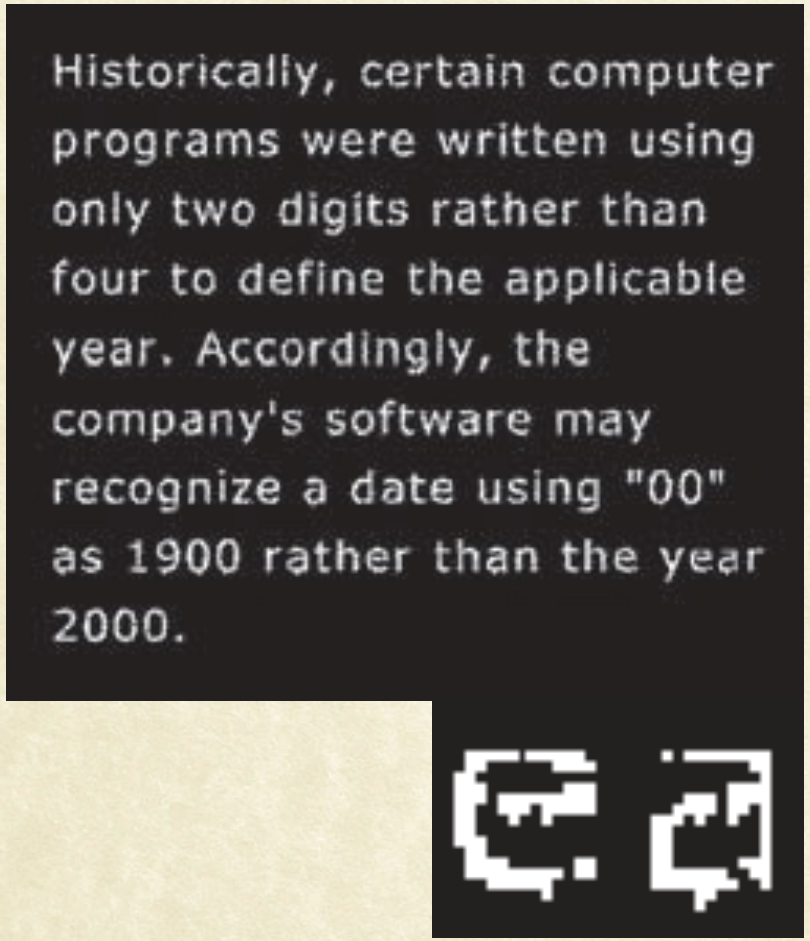
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0
0	0	0	0	1	1	1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

Dilation

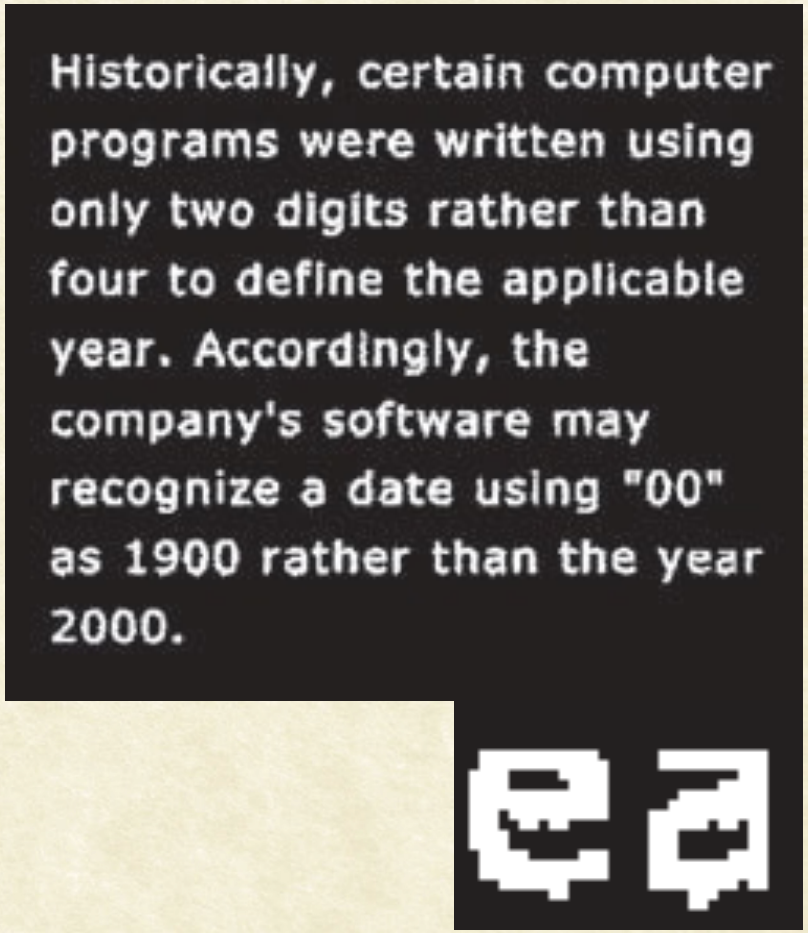
- Expands foreground objects
- Foreground holes are shrunk / removed
- Broken objects are connected
- NOTE: Multiple iterations of dilation



Dilation: Application



1	1	1
1	1	1
1	1	1



Opening and Closing

- We would like to maintain the size of objects while filling holes or removing connections
- Opening: Erode, then Dilate

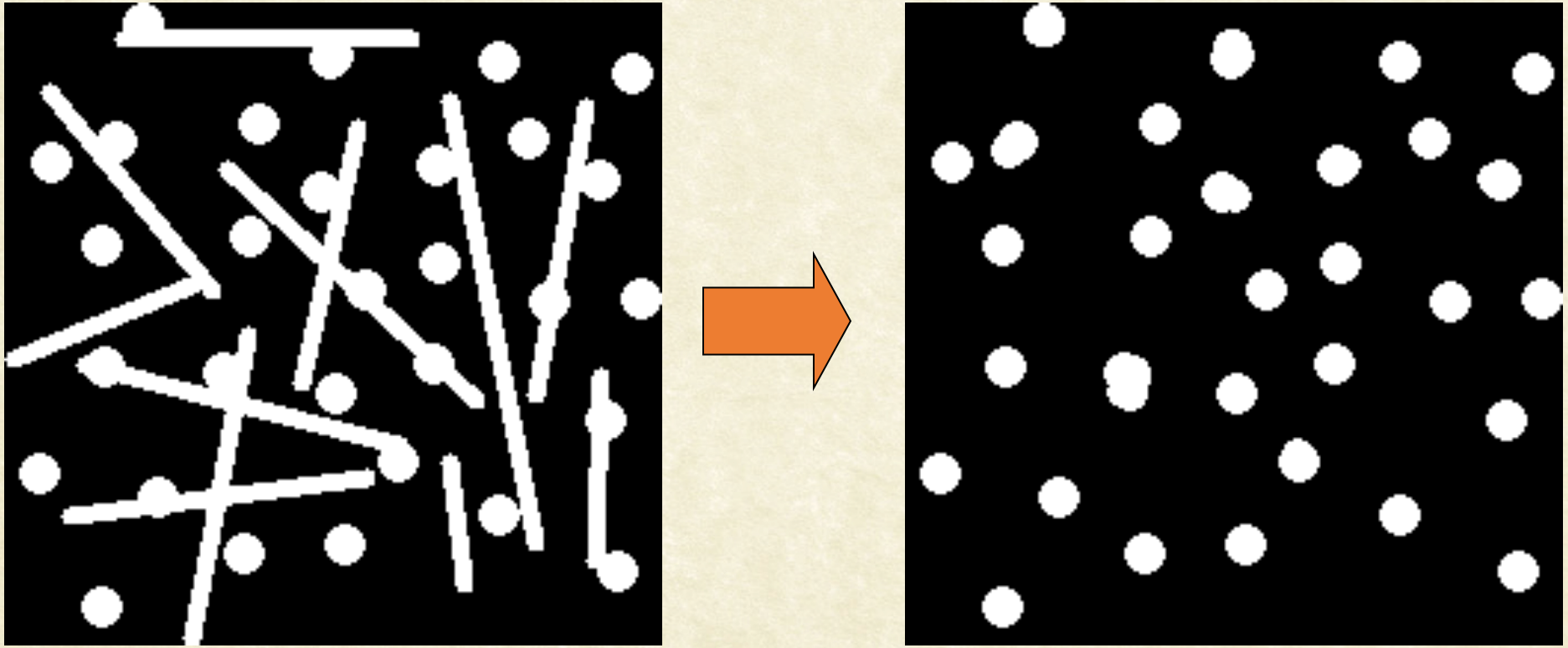
$$A \circ B = (A \ominus B) \oplus B$$

- Closing: Dilate, then Erode

$$A \cdot B = (A \oplus B) \ominus B$$

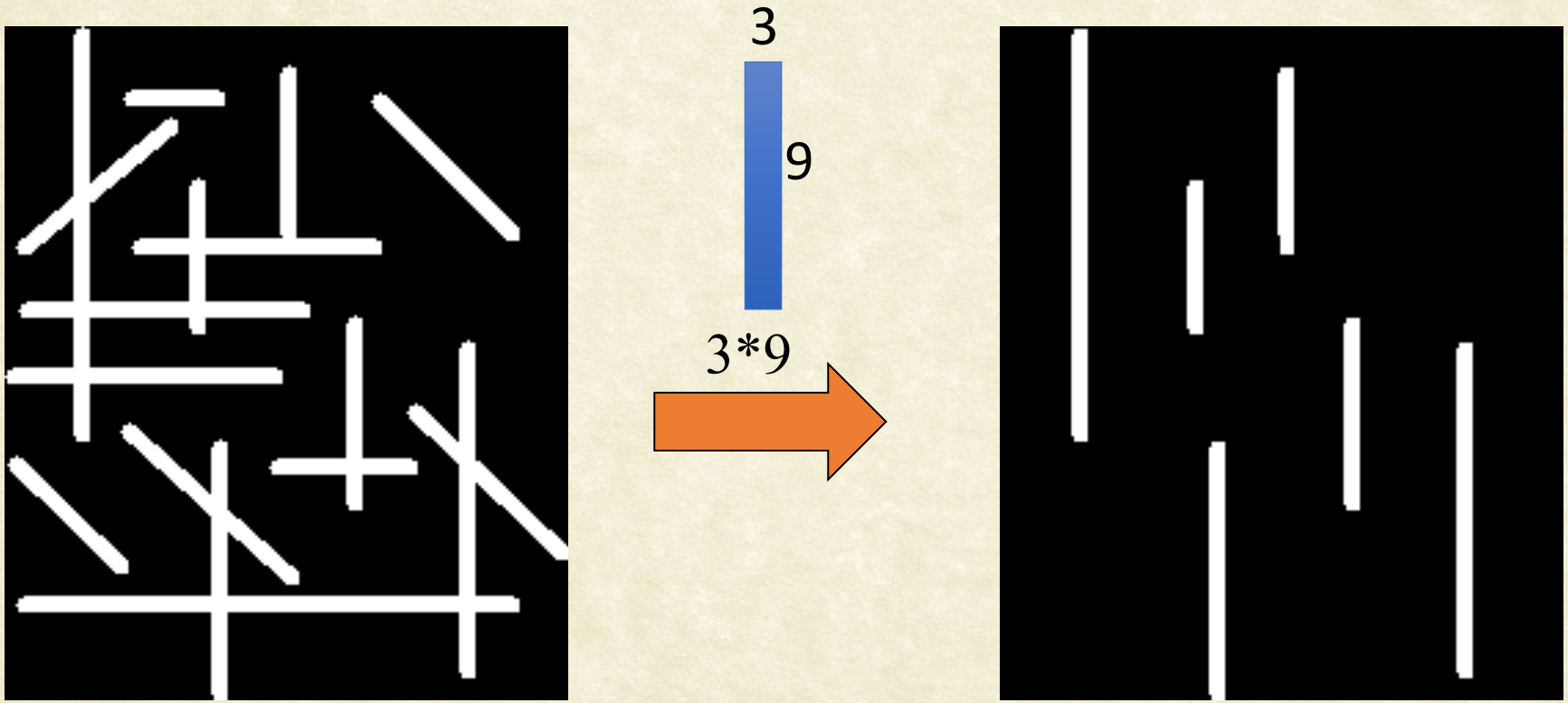
Opening: Example

- Opening with a 11 pixel diameter disc



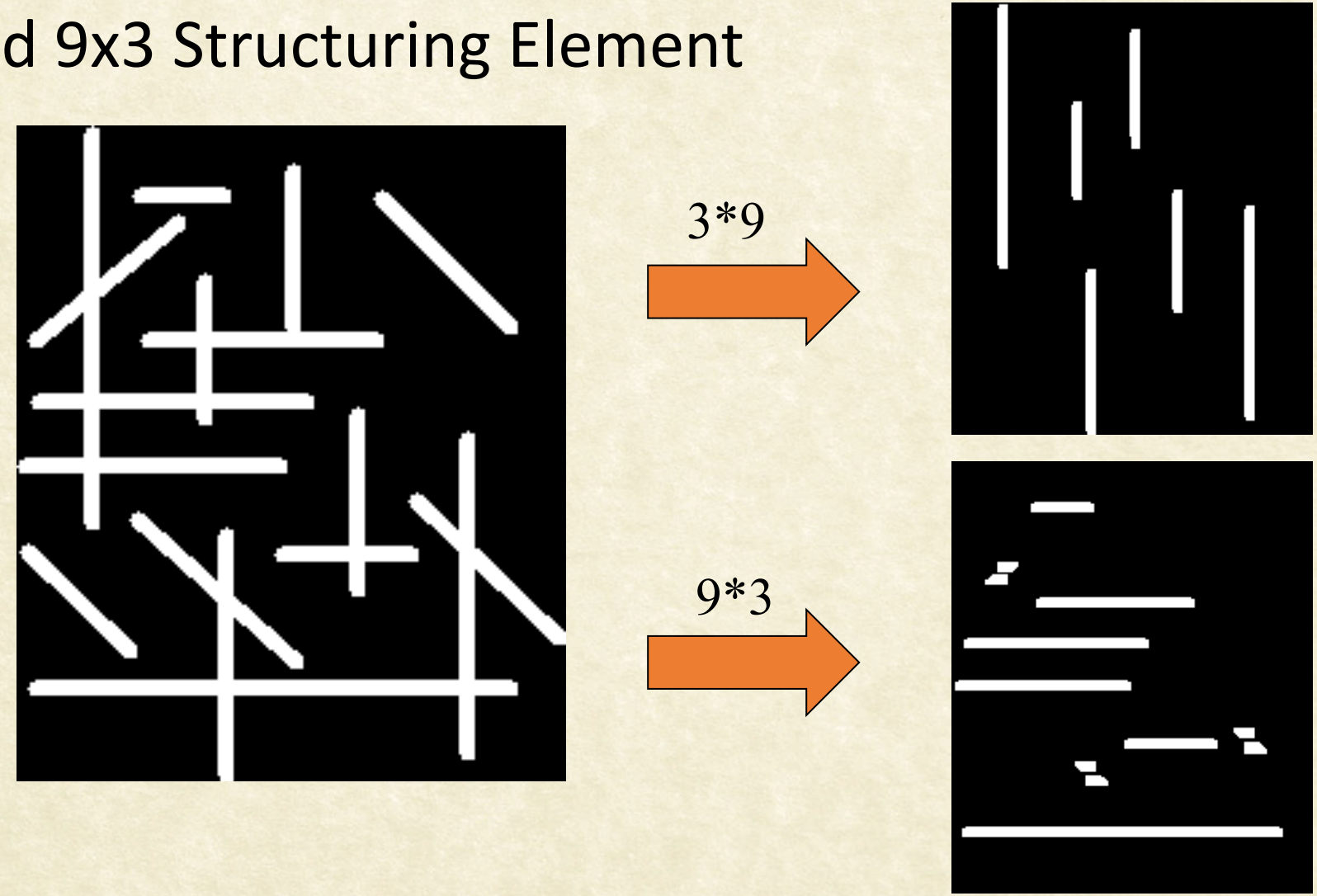
Opening: Another Example

- 3x9 Structuring Element



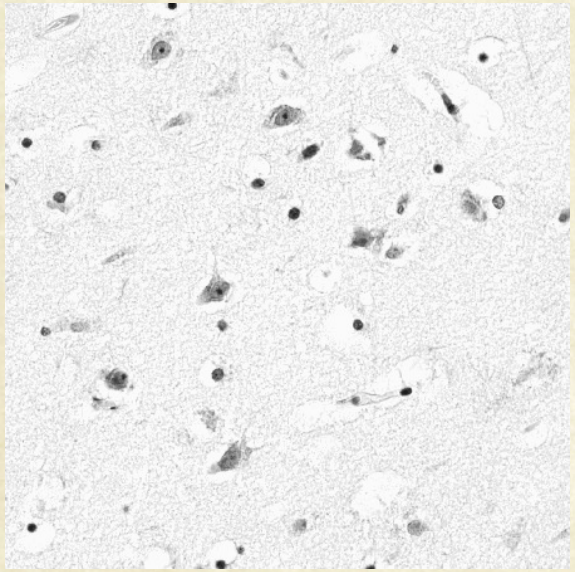
Opening: Another Example

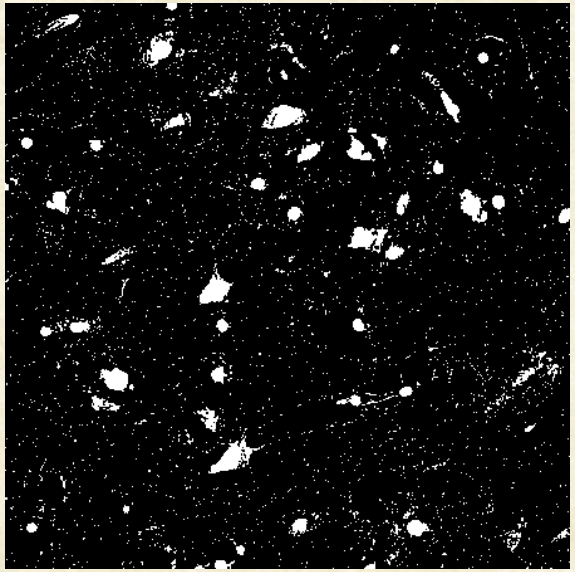
- 3x9 and 9x3 Structuring Element

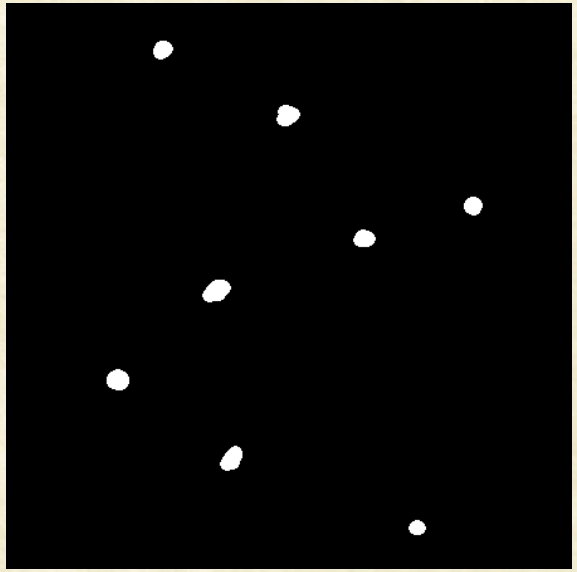


Use Opening for Separating Blobs

- Use large structuring element that fits into the big blobs
- Structuring Element: 11 pixel disc







Opening

- Similar to Erosion
  - Spot and noise removal
  - Less destructive
- Erosion followed by Dilation
- *Same structuring element for both operations.*
- Opening is **idempotent**
  - Repeated application has no further effects!

Closing (Dilation then Erosion)

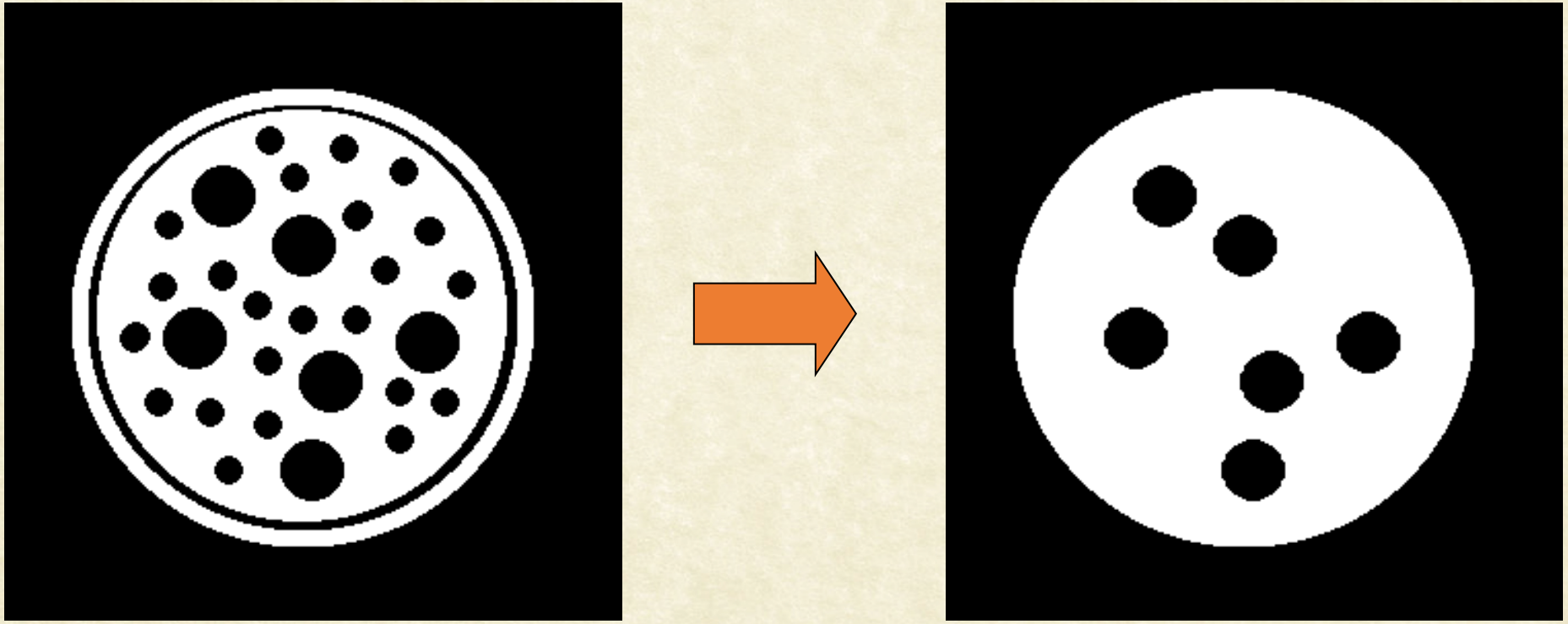
- Structuring element: 3x3 square

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

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

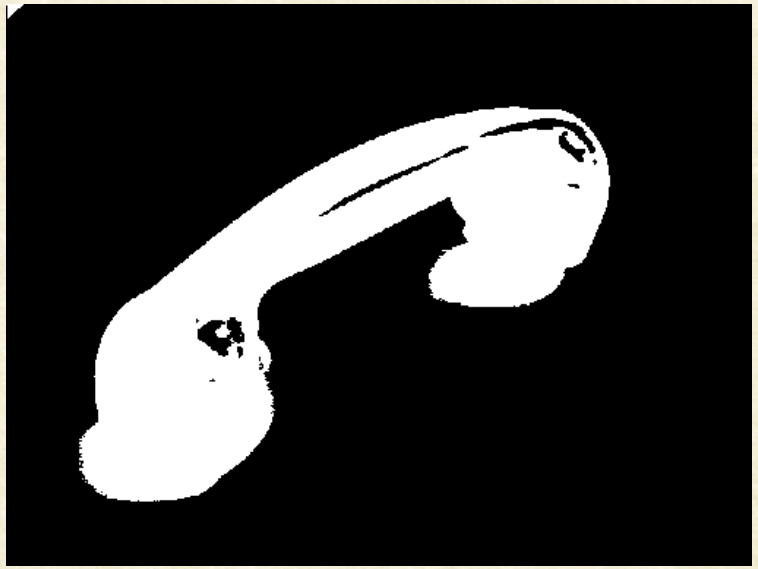
Closing: Example

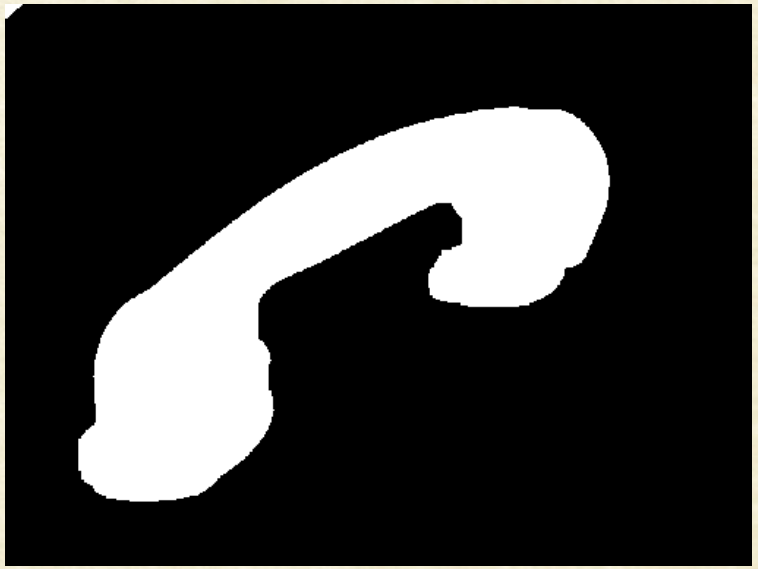
- Closing operation with a 22 pixel disc
- Closes small holes in the foreground



Closing Example

1. Threshold
2. Closing with disc of size 20



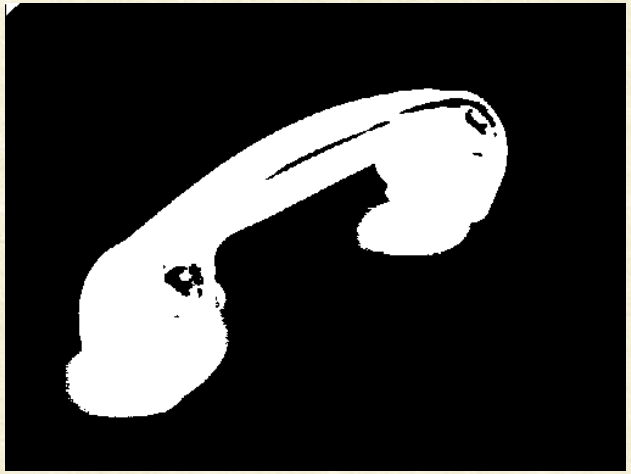


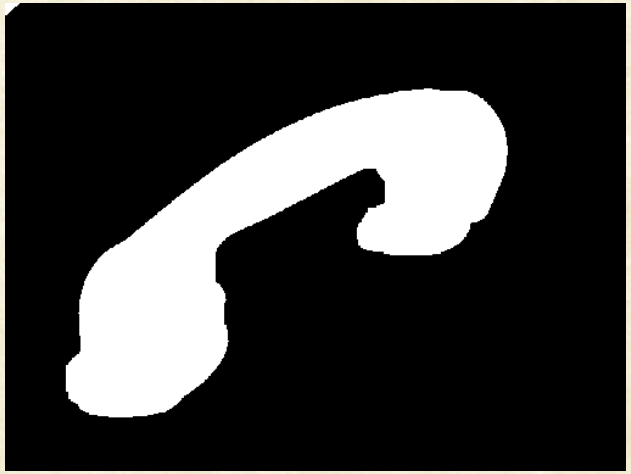
Thresholded

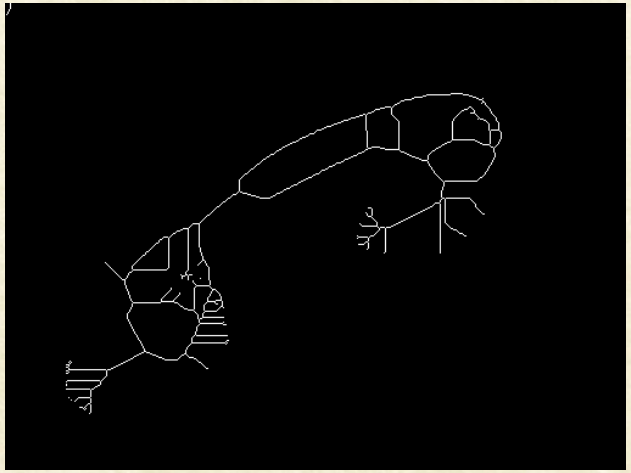
Closed

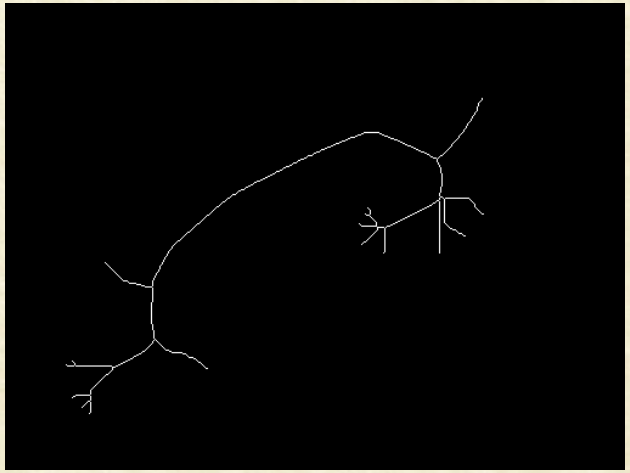
Application of Closing

- Good for further processing: E.g. Skeleton operation looks better for closed image!



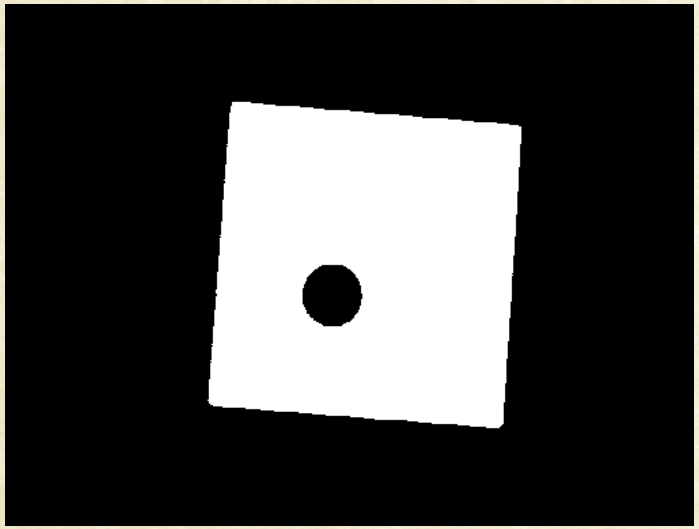
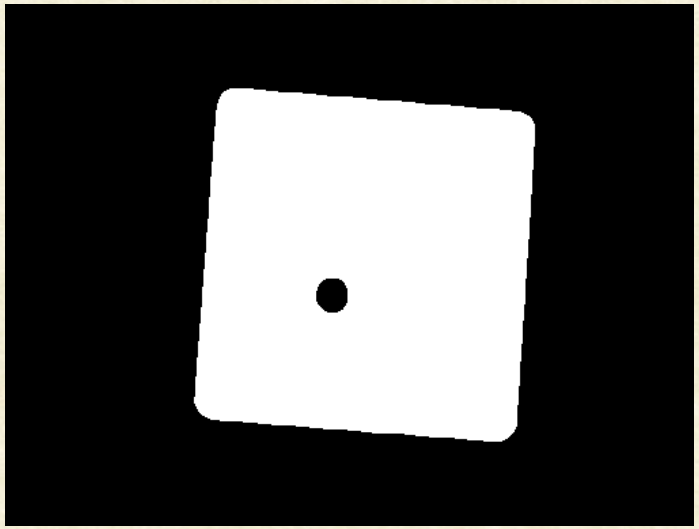
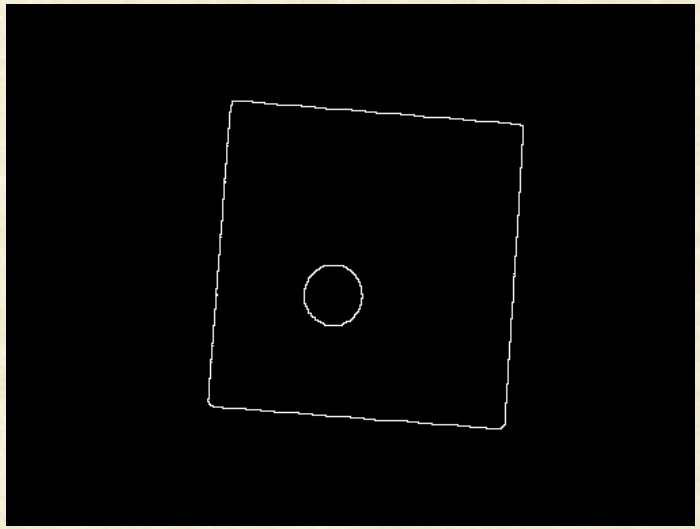






Boundary Detection

1. Dilate input image
2. Subtract input image from dilated image
3. Boundaries remain!

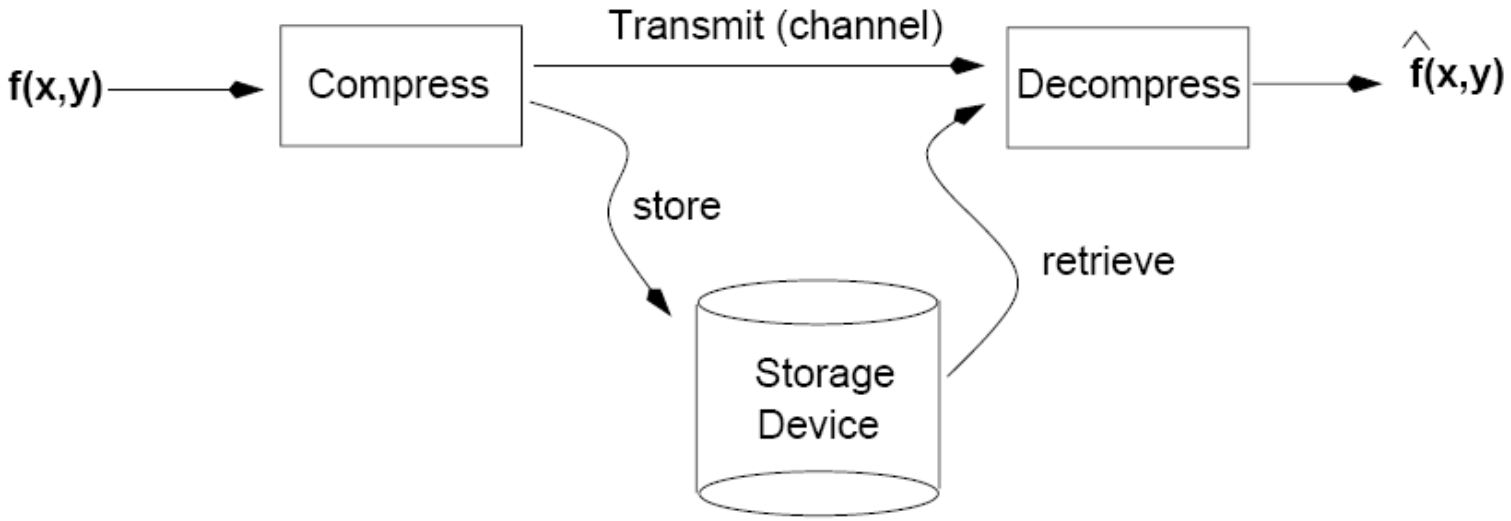




Can use erosion also ..

Fig 3: (a) Original Image (linkon.tif) (B) After erosion operation (C) Boundary Extraction with the help of Erosion.

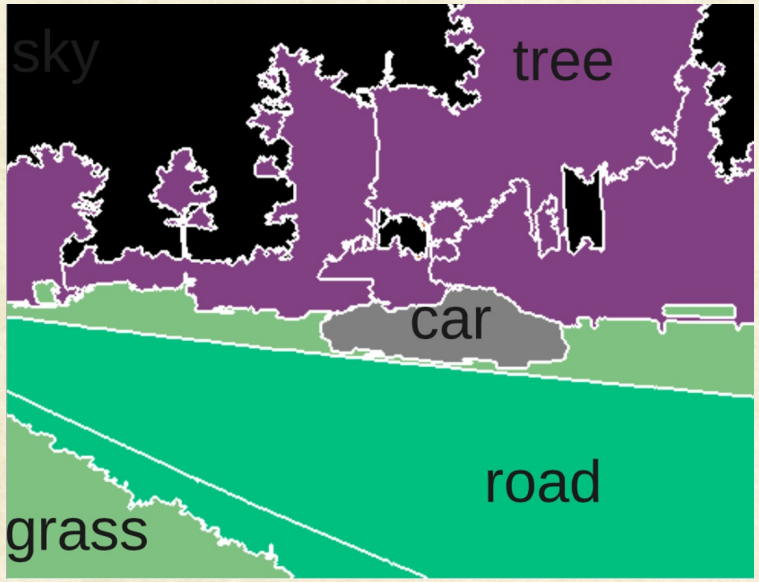
Image Compression

- Goal: Reduce amount of data required to represent a digital <u>image</u> (c.f. signal).



.. By exploiting redundancies in image data

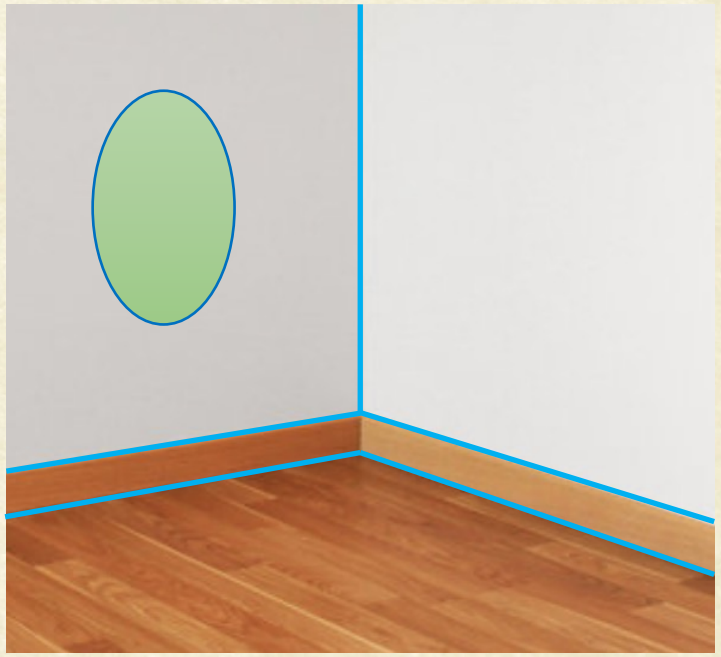
Urge to Group



- Group similar pixels together as objects.
- Group semantically meaningful pixels together as objects.
- Is appearance similarity the same as semantic similarity?

Harris Corner Detector

Detecting Interest Points

Hough Transform

Detecting Lines, Circles, etc.