DTU

Lazaros Nalpantidis

Image Processing

Outline

- What is Image Processing?

- Color
- Linear Filtering
- Non-linear Filters / Thresholding
- Morphology
- Connected Components Analysis

- Summary

Outline

- <u>What is Image Processing?</u>

- Color
- Linear Filtering
- Non-linear Filters / Thresholding
- Morphology
- Connected Components Analysis

- Summary

What is Image Processing?

- Image processing is:
  - the operations we perform on an image to change or enhance it and make it suitable for further analysis,
  - so that useful information can be highlighted or get extracted from it.

Outline

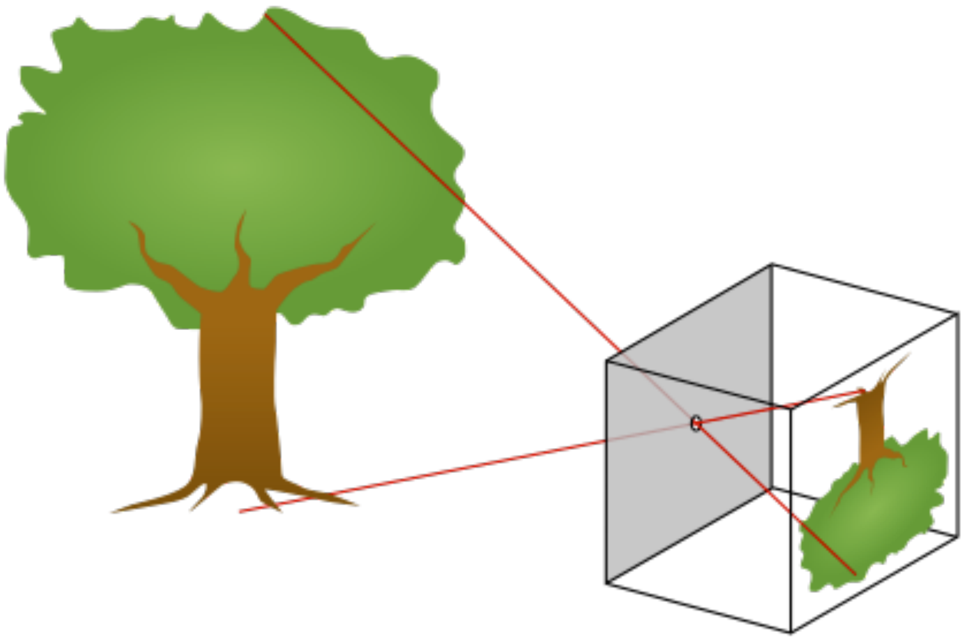
- What is Image Processing?

- <u>Color</u>
- Linear Filtering
- Non-linear Filters / Thresholding
- Morphology
- Connected Components Analysis

- Summary

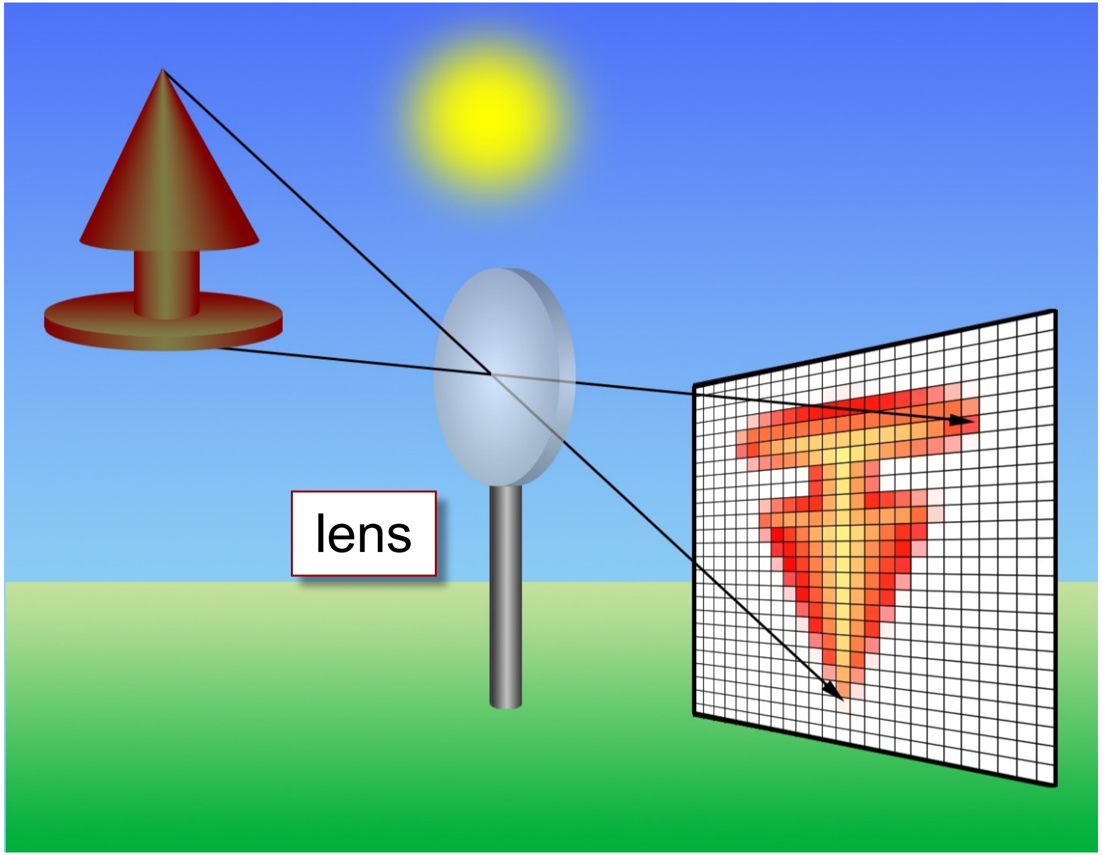
Image

- Pinhole model



Image

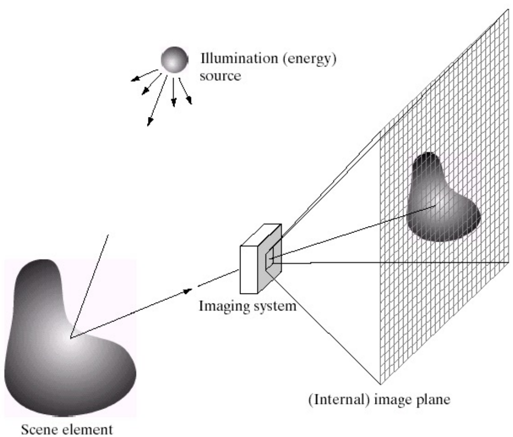
- Image Formation

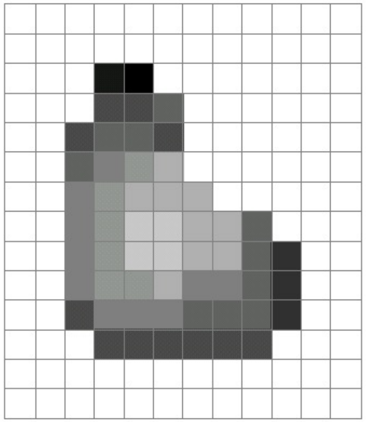


credtis: Richard Alan Peters II

Image

- What is a (digital) image?
  – an image is a matrix/table (descritized 2D space)
  – each cell can take discrete/integer values (quantized sample values)





255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	20	0	255	255	255	255	255	255	255
255	255	255	75	75	75	255	255	255	255	255	255
255	255	75	95	95	75	255	255	255	255	255	255
255	255	96	127	145	175	255	255	255	255	255	255
255	255	127	145	175	175	175	255	255	255	255	255
255	255	127	145	200	200	175	175	95	255	255	255
255	255	127	145	200	200	175	175	95	47	255	255
255	255	127	145	145	175	127	127	95	47	255	255
255	255	74	127	127	127	95	95	95	47	255	255
255	255	255	74	74	74	74	74	74	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255

Image

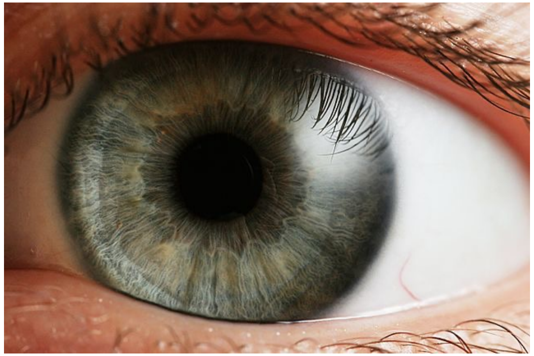
- What is a (digital) image?
  - an image is a matrix/table (descritized 2D space)
  - each cell can take discrete/integer values (quantized sample values)

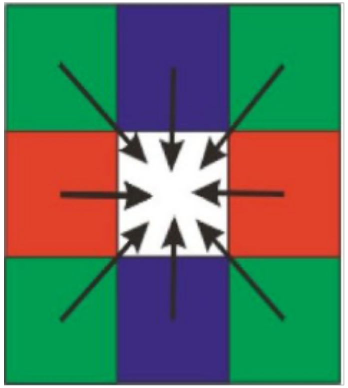
j

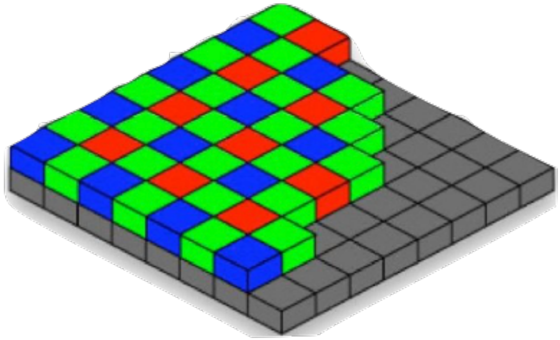
121	121	118	111	. . .	21
134	136	137	132	. . .	23
133	131	136	136	. . .	25
136	145	148	151	. . .	34
137	140	147	149	. . .	54
. . .	. . .	. . .	. . .	. . .	. . .
231	233	243	244	. . .	179

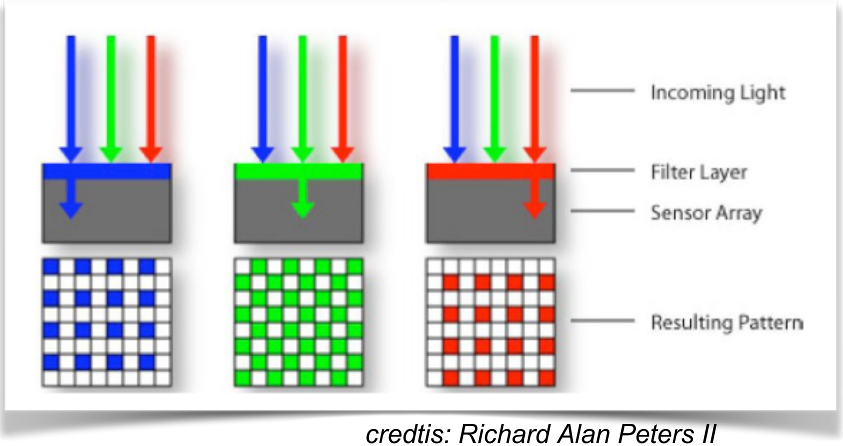
Color

- Color Images
  - Bayer Color Filter Array





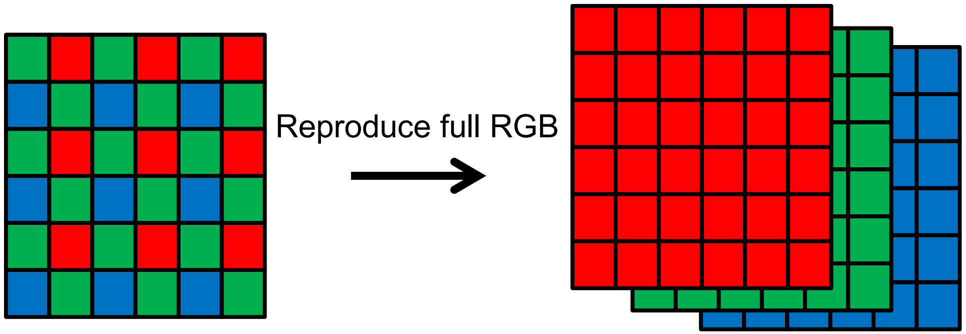


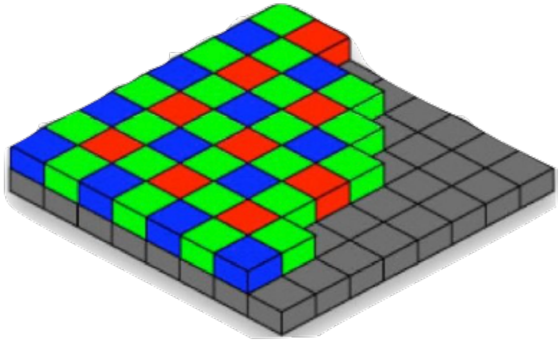


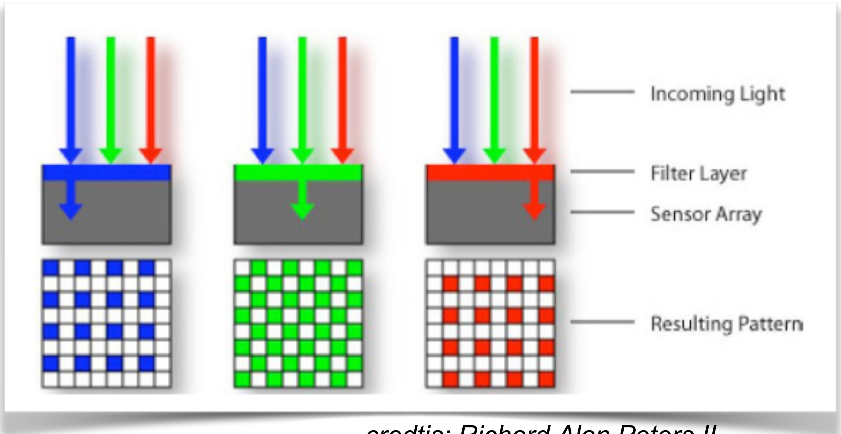
credtis: Richard Alan Peters II

Color

- Color Images
  - Bayer Color Filter Array



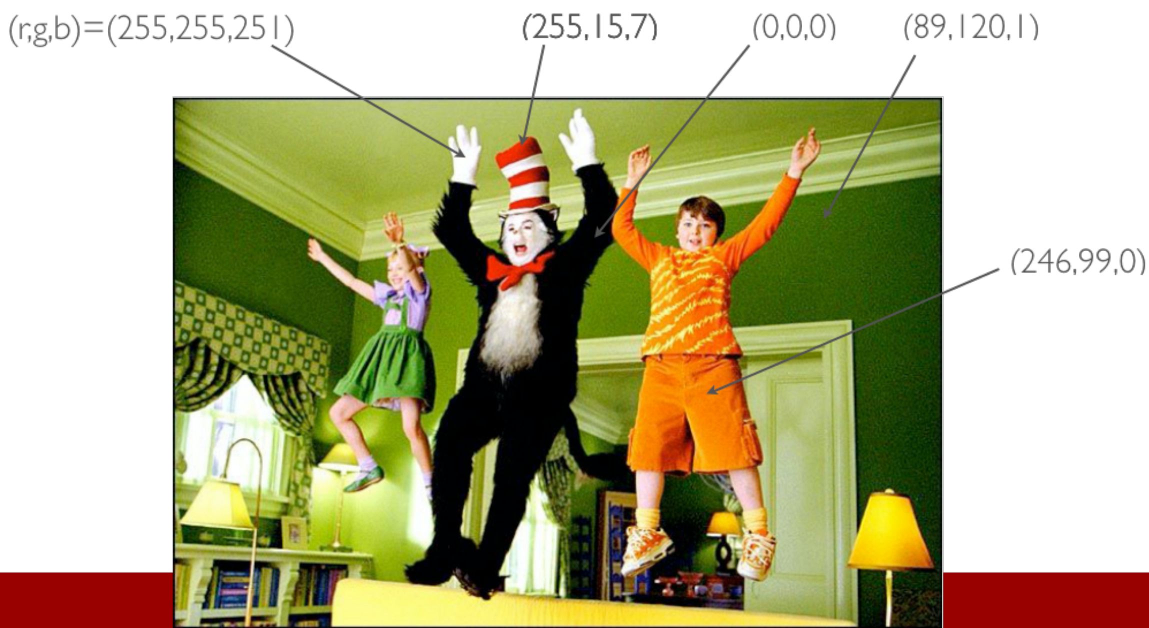




credtis: Richard Alan Peters II

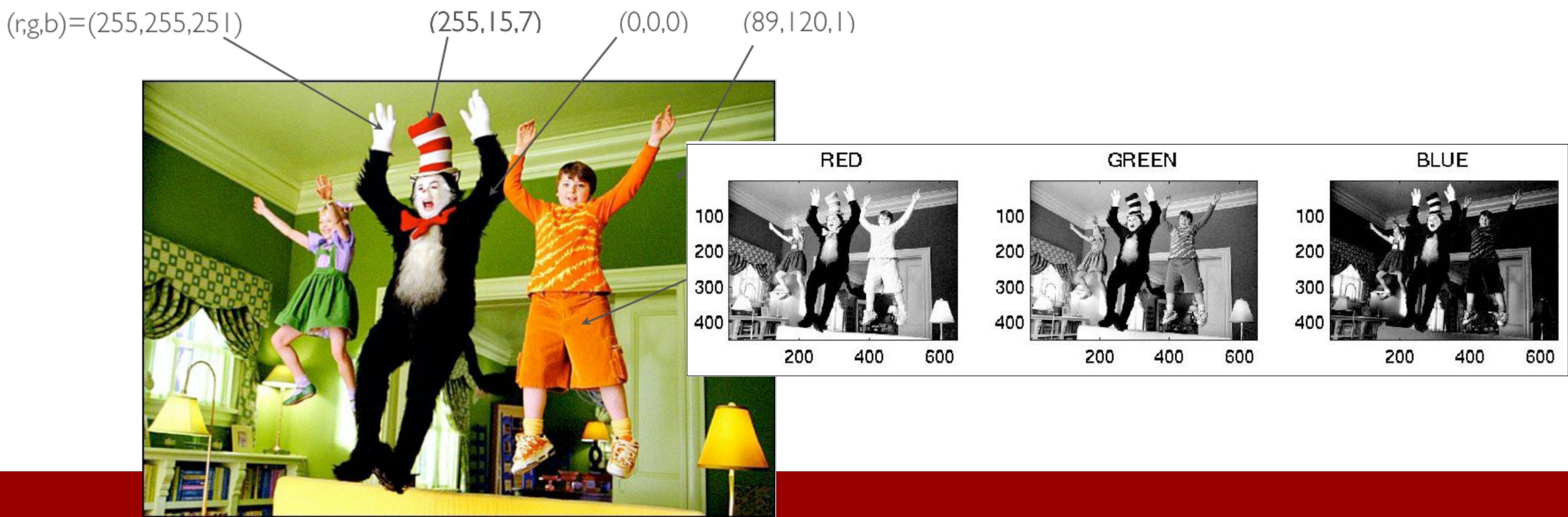
Color

- Color Images
  - are composed of 3 2D images: R(x,y), G(x,y), B(x,y)
  - Each pixel (x,y) of these 3 images consists of values between 0 and 255
  - The values of a specific pixel (x_1,y_1) in the 3 images R, G, B describe the red-ness, green-ness and blue-ness of that particular pixel.



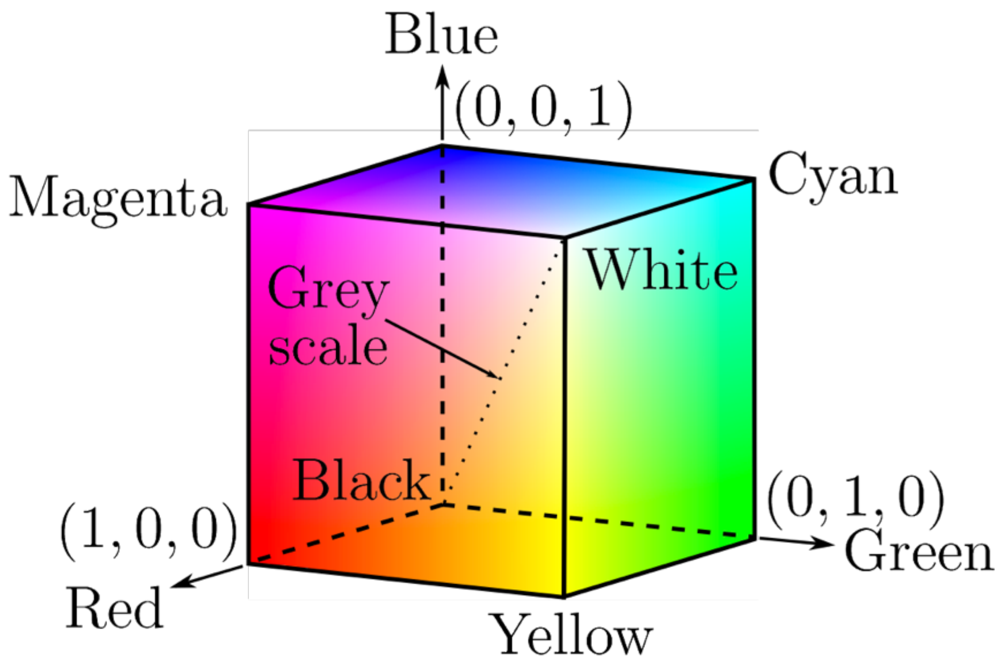
Color

- Color Images
  - are composed of 3 2D images: R(x,y), G(x,y), B(x,y)
  - Each pixel (x,y) of these 3 images consists of values between 0 and 255
  - The values of a specific pixel (x_1,y_1) in the 3 images R, G, B describe the red-ness, green-ness and blue-ness of that particular pixel.



Color

- RGB Color Space
  - – all colors can be reproduced by mixing Red, Green and Blue



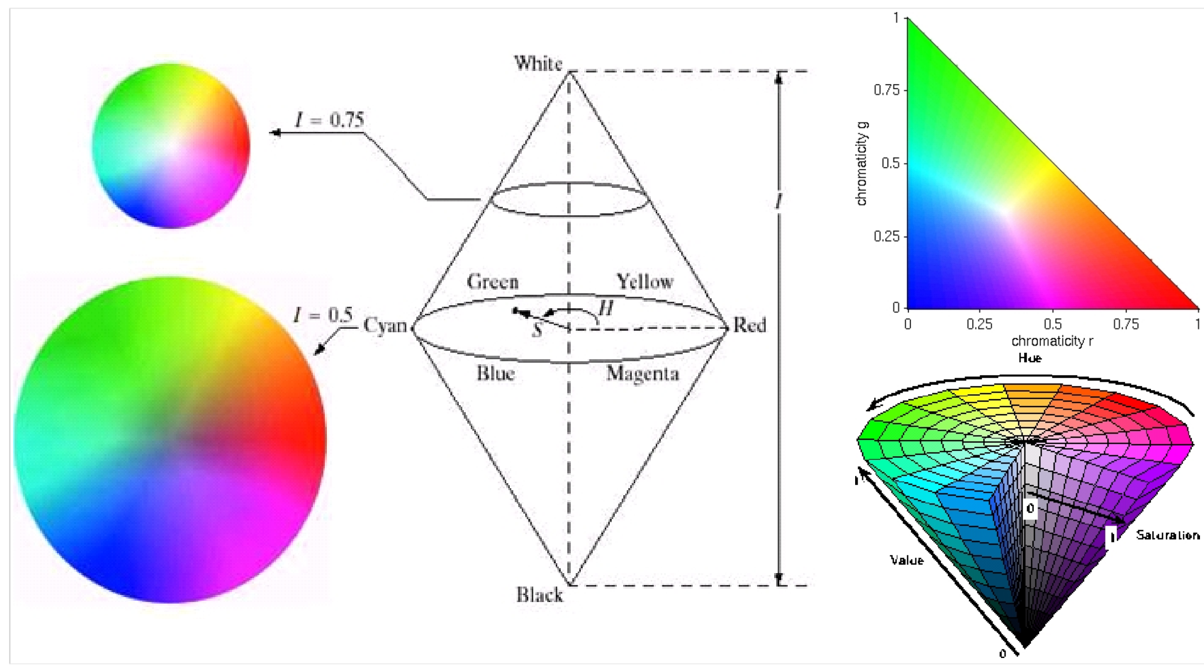
Color

- Many other Color Spaces existe.g. L*a*b* color space
- or HSI/HSV Color spaces

- **Hue**: color, chromatic information

- **Saturation**: purity, amount of white

- **Intensity**:



Outline

- What is Image Processing?

- Color
- Linear Filtering
- Non-linear Filters / Thresholding
- Morphology
- Connected Components Analysis

- Summary

Linear Filtering

- What are Linear Filters in Image Processing?
- What are they used for?

Noise reduction

- Nearby pixels are likely to belong to same object
  – thus likely to have similar color
- Replace each pixel by *average of neighbors*

Mean filtering

0	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	0
0	0	10	20	20	20	10	40	0	0
0	10	20	30	0	20	10	0	0	0
0	10	0	30	40	30	20	10	0	0
0	10	20	30	40	30	20	10	0	0
0	10	20	10	40	30	20	10	0	0
0	10	20	30	30	20	10	0	0	0
0	0	10	20	20	0	10	0	20	0
0	0	0	10	10	10	0	0	0	0

$$(0 + 0 + 0 + 10 + 40 + 0 + 10 + 0 + 0)/9 = 6.66$$

Slide from: Bharath Hariharan

Mean filtering

0	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	0
0	0	10	20	20	20	10	40	0	0
0	10	20	30	0	20	10	0	0	0
0	10	0	30	40	30	20	10	0	0
0	10	20	30	40	30	20	10	0	0
0	10	20	10	40	30	20	10	0	0
0	10	20	30	30	20	10	0	0	0
0	0	10	20	20	0	10	0	20	0
0	0	0	10	10	10	0	0	0	0

$$(0 + 0 + 0 + 0 + 0 + 10 + 0 + 0 + 0 + 0 + 20 + 10 + 40 + 0 + 0 + 20 + 10 + 0 + 0 + 0 + 30 + 20 + 10 + 0 + 0)/25 = 6.8$$

Mean filtering

0	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	0
0	0	10	20	20	20	10	40	0	0
0	10	20	30	0	20	10	0	0	0
0	10	0	30	40	30	20	10	0	0
0	10	20	30	40	30	20	10	0	0
0	10	20	10	40	30	20	10	0	0
0	10	20	30	30	20	10	0	0	0
0	0	10	20	20	0	10	0	20	0
0	0	0	10	10	10	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	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	0

$$(0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 10)/9 = 1.11$$

Slide from: Bharath Hariharan

Mean filtering

0	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	0
0	0	10	20	20	20	10	40	0	0
0	10	20	30	0	20	10	0	0	0
0	10	0	30	40	30	20	10	0	0
0	10	20	30	40	30	20	10	0	0
0	10	20	10	40	30	20	10	0	0
0	10	20	30	30	20	10	0	0	0
0	0	10	20	20	0	10	0	20	0
0	0	0	10	10	10	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	1	4	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	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$(0 + 0 + 0 + 0 + 0 + 10 + 0 + 10 + 20)/9 = 4.44$$

Slide from: Bharath Hariharan

Mean filtering

0	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	0
0	0	10	20	20	20	10	40	0	0
0	10	20	30	0	20	10	0	0	0
0	10	0	30	40	30	20	10	0	0
0	10	20	30	40	30	20	10	0	0
0	10	20	10	40	30	20	10	0	0
0	10	20	30	30	20	10	0	0	0
0	0	10	20	20	0	10	0	20	0
0	0	0	10	10	10	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	1	4	8	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	0	0	0
0	0	0	0	0	0	0	0	0	0

$$(0 + 0 + 0 + 0 + 10 + 10 + 10 + 20 + 20)/9 = 7.77$$

Slide from: Bharath Hariharan

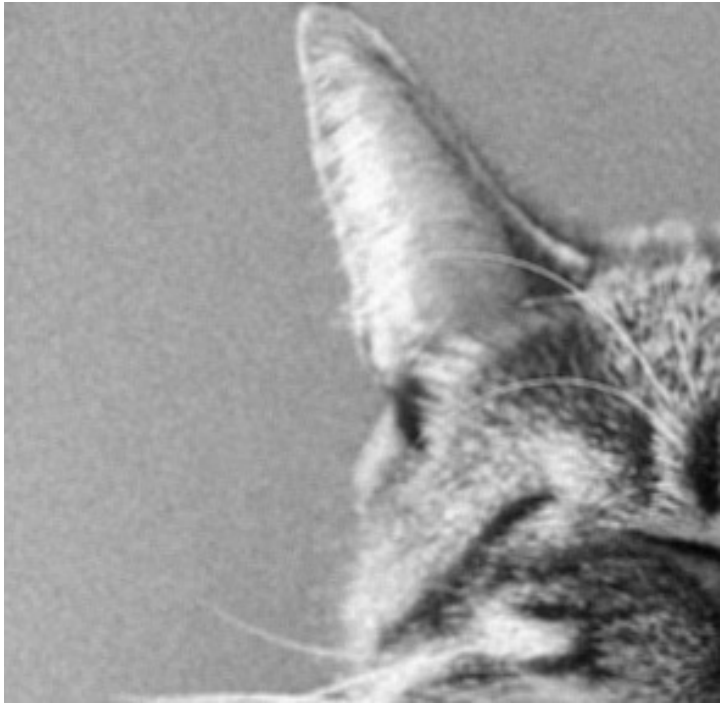
Mean filtering

0	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	0
0	0	10	20	20	20	10	40	0	0
0	10	20	30	0	20	10	0	0	0
0	10	0	30	40	30	20	10	0	0
0	10	20	30	40	30	20	10	0	0
0	10	20	10	40	30	20	10	0	0
0	10	20	30	30	20	10	0	0	0
0	0	10	20	20	0	10	0	20	0
0	0	0	10	10	10	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	1	4	8	10	8	9	6	4	0
0	4	11	13	16	11	12	7	4	0
0	6	14	19	23	19	18	10	6	0
0	8	18	23	28	23	17	8	2	0
0	8	16	26	31	30	20	10	3	0
0	10	18	27	29	27	17	8	2	0
0	8	14	22	22	20	11	8	3	0
0	4	11	17	17	12	6	4	2	0
0	0	0	0	0	0	0	0	0	0

Slide from: Bharath Hariharan

Noise reduction using mean filtering



Slide from: Bharath Hariharan

Filters

- Filtering
  - Form a new image whose pixels are a combination of the original pixels
- Why?
  - To get useful information from images
    - E.g., extract edges or contours (to understand shape)
  - To enhance the image
    - E.g., to blur to remove noise
    - E.g., to sharpen to “enhance image” a la CSI

Slide from: Bharath Hariharan

Mean filtering

- Replace pixel by mean of neighborhood

10	5	3
4	5	1
1	1	7

Local image data

f

	4.1	

Modified image data

S[*f*]

$$S[f](m,n) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(m+i, n+j)/9$$

A more general version

10	5	3
4	5	1
1	1	7

Local image data

	7	

Kernel / filter

$$S[f](m,n) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} \overbrace{w(i,j)}^{\text{Kernel / filter}} f(m+i, n+j)$$

A more general version

0	10	5	7	0
5	11	6	8	3
9	22	4	5	1
2	9	14	6	7
3	10	15	12	9

Local image data

		7		

Kernel size = 2k+1

$$S[f](m,n) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w(i,j) f(m+i, n+j)$$

A more general version

$$S[f](m,n) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w(i,j) f(m+i, n+j)$$

- w(i,j) = $1/(2k+1)^2$ for mean filter
- If w(i,j) >=0 and sum to 1, *weighted mean*
- But w(i,j) can be *arbitrary real numbers!*

Convolution and cross-correlation

- Cross correlation

$$S[f] = w \otimes f$$

$$S[f](m,n) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w(i,j) f(m+i, n+j)$$

- Convolution

$$S[f] = w * f$$

$$S[f](m,n) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w(i,j) f(m-i, n-j)$$

Slide from: Bharath Hariharan

Cross-correlation

1	2	3
4	5	6
7	8	9

w

1	2	3
4	5	6
7	8	9

f

1*1 + 2*2 + 3*3 + 4*4 + 5*5 + 6*6 + 7*7 + 8*8 + 9*9

Slide from: Bharath Hariharan

Convolution

1	2	3
4	5	6
7	8	9

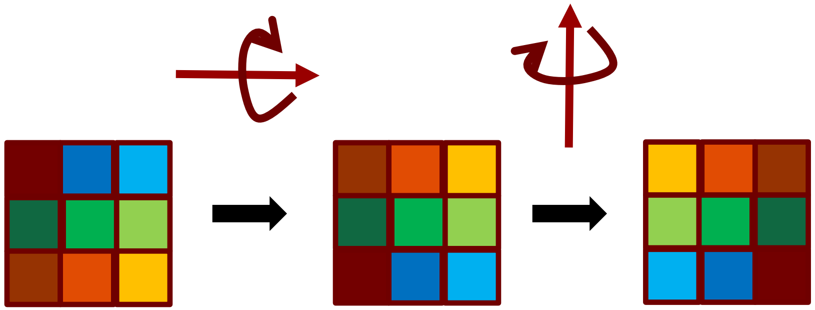
w

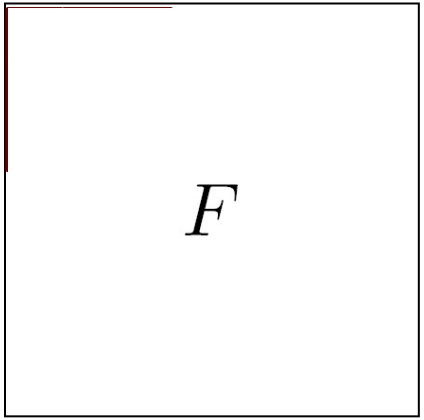
1	2	3
4	5	6
7	8	9

f

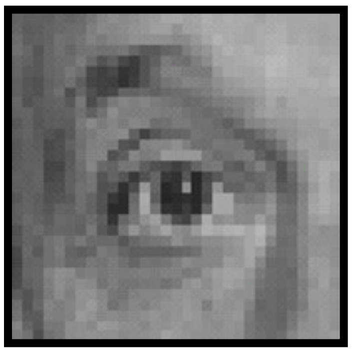
1*9 + 2*8 + 3*7 + 4*6 + 5*5 + 6*4 + 7*3 + 8*2 + 9*1

Convolution





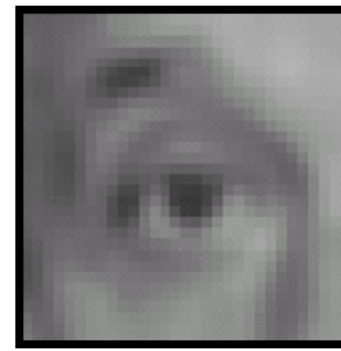
Filters: examples



Original (f)

$$* \; \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \; =$$

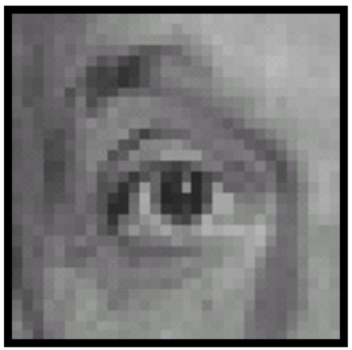
Kernel (k)



Blur (with a mean filter) (g)

Source: D. Lowe

Filters: examples



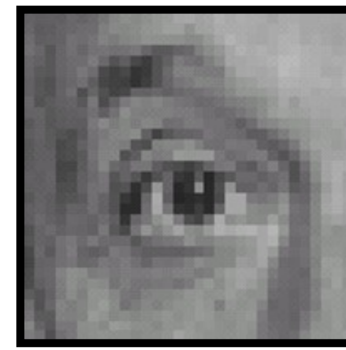
Original (f)

$*$

0	0	0
0	1	0
0	0	0

Kernel (k)

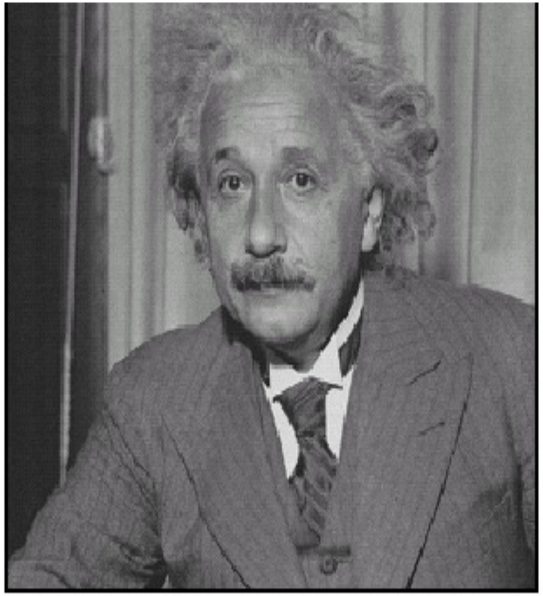
$=$



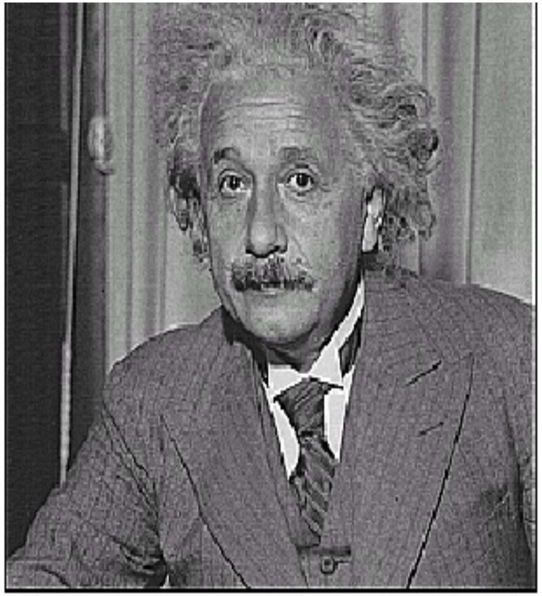
Identical image (g)

Source: D. Lowe

Sharpening



before

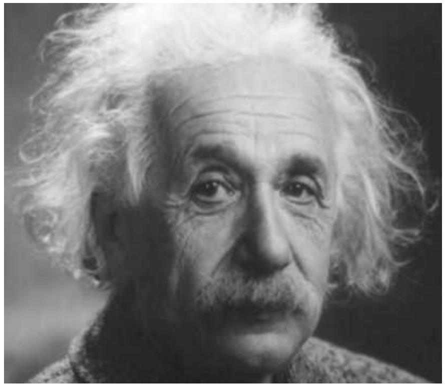


after

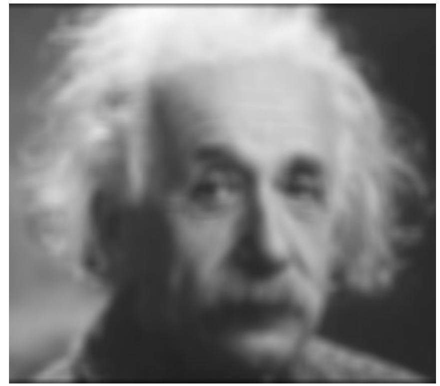
Source: D. Lowe

Sharpening

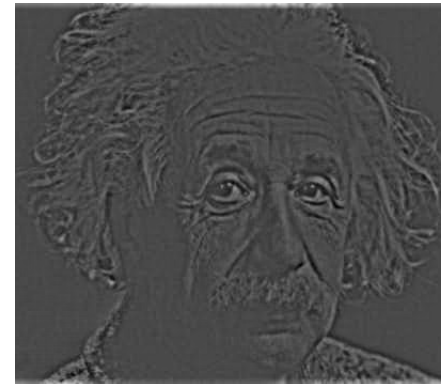
- What does blurring take away?



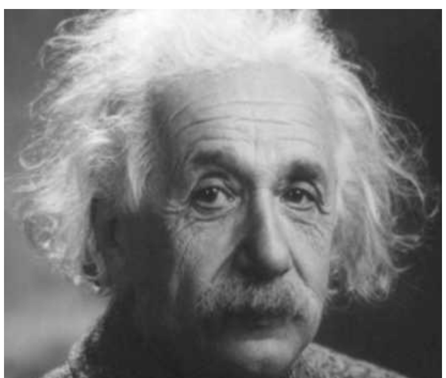
–



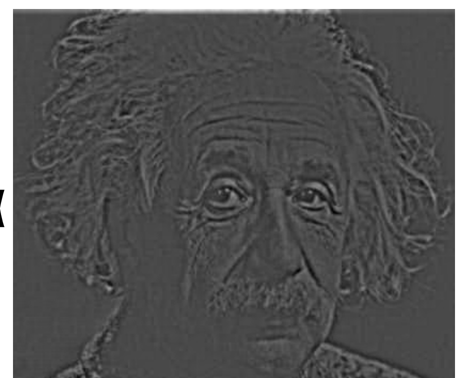
=



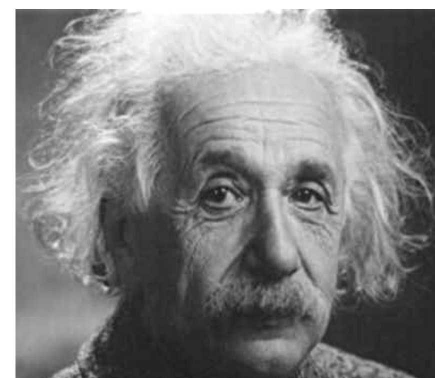
Let’s add it back:



+ α

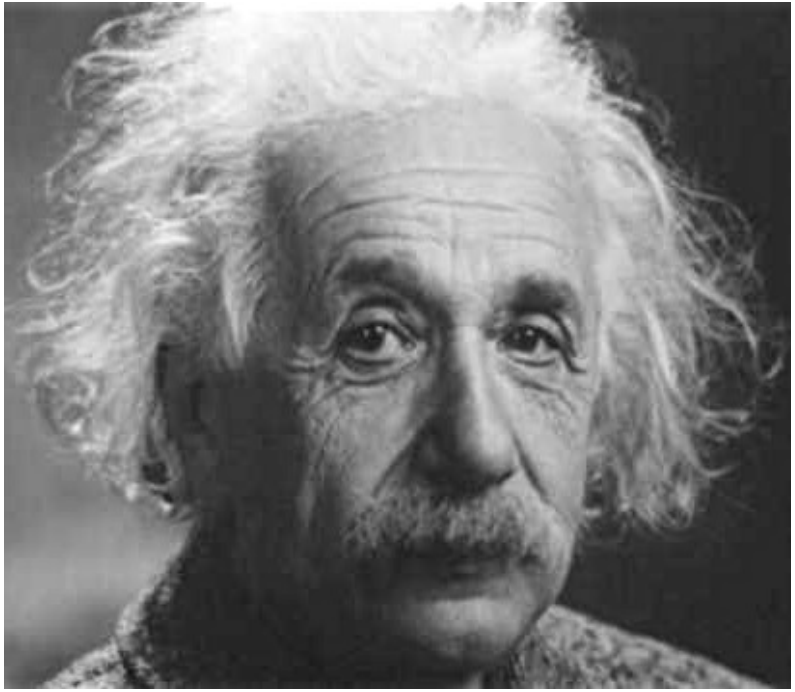


=



Slide from: Bharath Hariharan

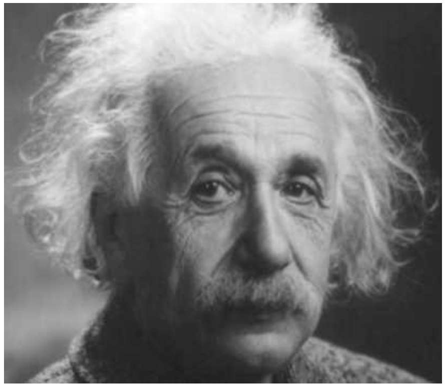
Sharpening



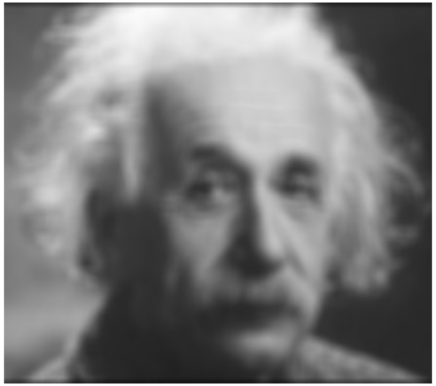
Slide from: Bharath Hariharan

Sharpening

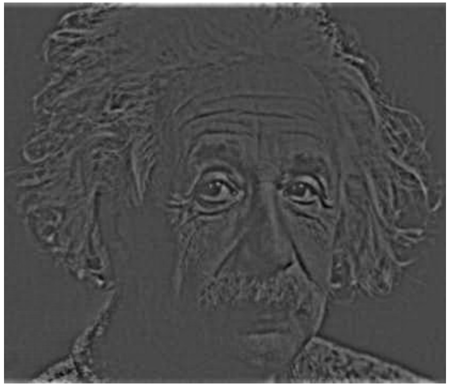
- What does blurring take away?



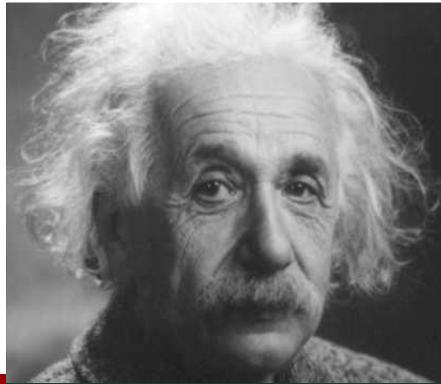
–



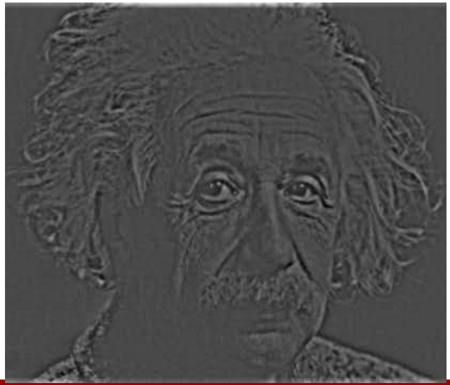
=



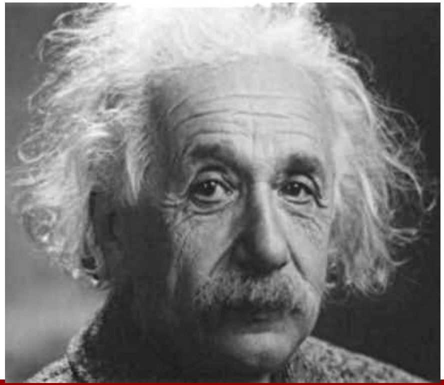
Let's add it back:



+ α



=



Sharpening

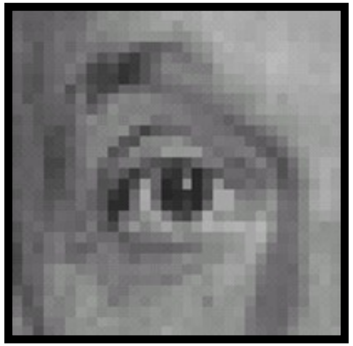
$$
\begin{aligned}
f_{sharp} &= f + \alpha(f - f_{blur}) \\
&= (1+\alpha)f - \alpha f_{blur} \\
&= (1+\alpha)(w * f) - \alpha(v * f)
\end{aligned}
$$

$$
\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}
$$

$$
= ((1+\alpha)w - \alpha v) * f
$$

Slide from: Bharath Hariharan

Sharpening filter



Original

$$* \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right) =$$

Sharpening filter
(accentuates edges)

Source: D. Lowe

Outline

- What is Image Processing?

- Color
- Linear Filtering
- <u>Non-linear Filters / Thresholding</u>
- Morphology
- Connected Components Analysis

- Summary

Non-linear filters: Thresholding

$$g(m,n) = \begin{cases} 255, & f(m,n) > A \\ 0 & otherwise \end{cases}$$

Non-linear filters: Thresholding

- What if Threshold could be adaptive (instead of a pre-defined value)?
  – Otsu’s method performs automatic image thresholding
    - The algorithm exhaustively searches for the threshold that minimizes the intra-class variance, defined as a weighted sum of variances of the two classes

Source: https://en.wikipedia.org/wiki/Otsu%27s_method

Non-linear filters: Rectification

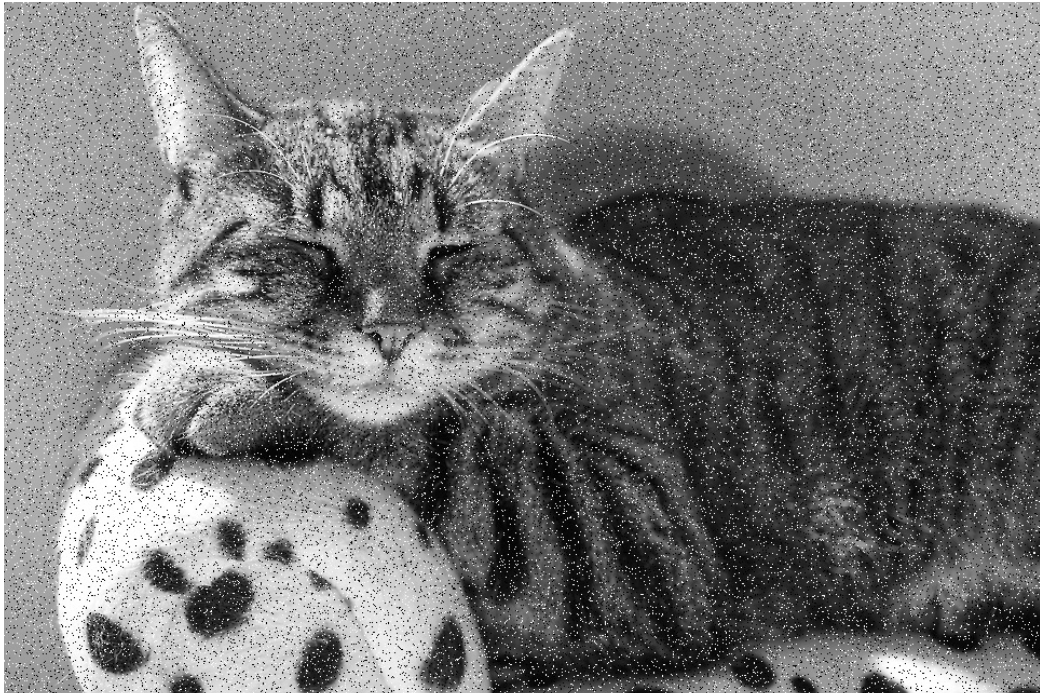
- $g(m,n) = \max(f(m,n), 0)$

- Crucial component of modern convolutional networks

Slide from: Bharath Hariharan

Non-linear filters

- Sometimes mean filtering does not work



Non-linear filters

- Sometimes mean filtering does not work

Slide from: Bharath Hariharan

Non-linear filters

- Mean is sensitive to outliers
- Median filter: Replace pixel by *median* of neighbors

Slide from: Bharath Hariharan

Non-linear filters

Slide from: Bharath Hariharan

Outline

- What is Image Processing?

- Color
- Linear Filtering
- Non-linear Filters / Thresholding
- <u>Morphology</u>
- Connected Components Analysis

- Summary

Morphology

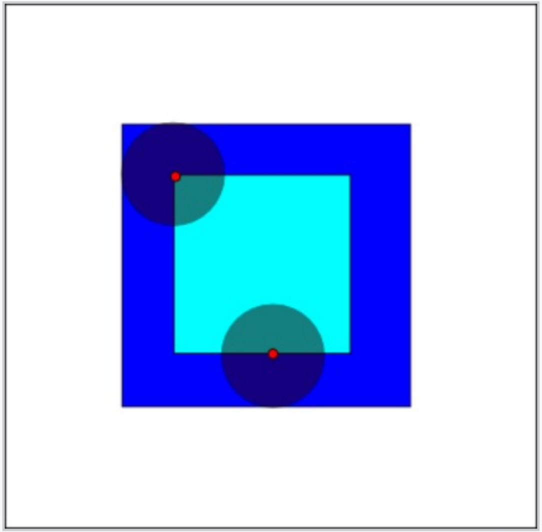
- The most common binary image operations are called Morphological Operations

- Basic Morphological Operations:
  - Erosion
  - Dilation
  - Opening
  - Closing
    - – All of them rely on convolution with a Structuring Element
      - » The Structuring Element can be a disk, rectangle, or of any other shape.

- There are also Grayscale Morphological Operations, apart from Binary ones.

Morphology

- The most common binary image operations are called Morphological Operations
- Basic Morphological Operations:
    - Erosion



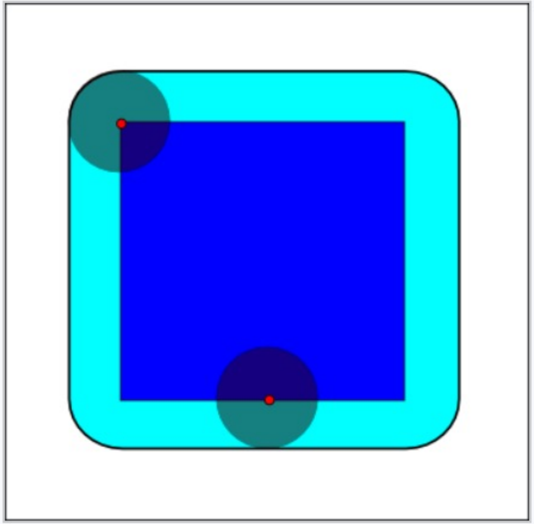
The erosion of the dark-blue square by a disk, resulting in the light-blue square.

Source:https://en.wikipedia.org/wiki/Mathematical_morphology

Morphology

- The most common binary image operations are called Morphological Operations

- Basic Morphological Operations:
  - Dilation



The dilation of the dark-blue square by a disk, resulting in the light-blue square with rounded corners.

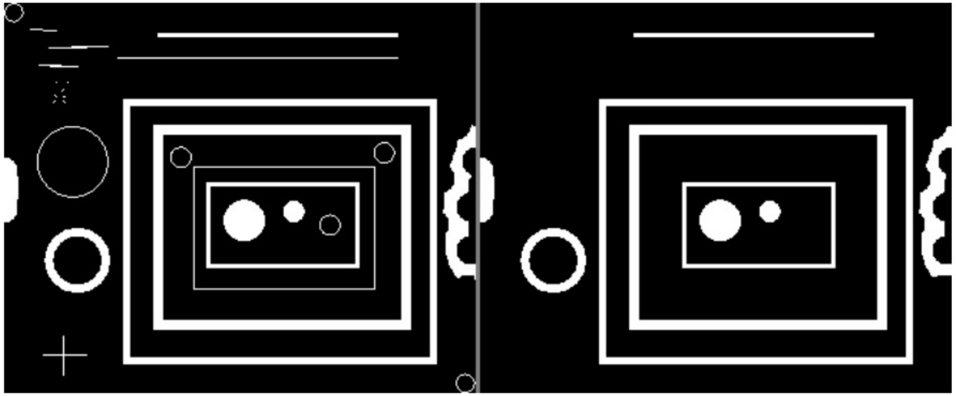
Source:https://en.wikipedia.org/wiki/Mathematical_morphology

Morphology

- The most common binary image operations are called Morphological Operations

- Basic Morphological Operations:
  - Opening = Erosion and then Dilation

    » Morphological opening is useful for removing small objects from an image while preserving the shape and size of larger objects in the image.



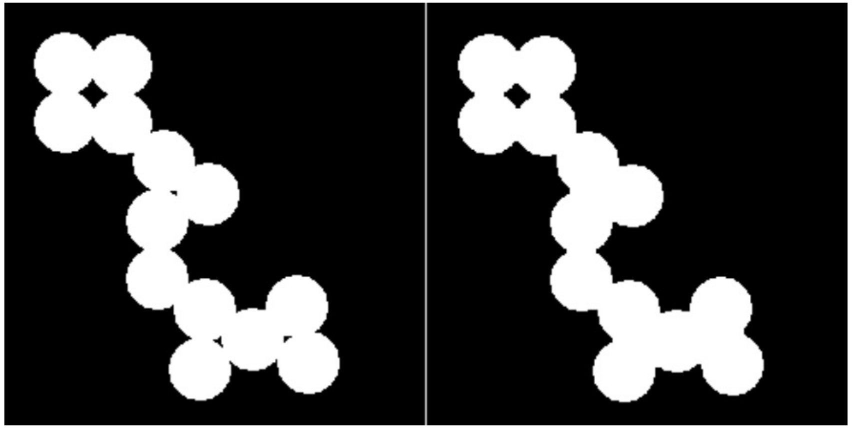
Source:https://se.mathworks.com/help/images/morphological-dilation-and-erosion.html

Morphology

- The most common binary image operations are called Morphological Operations

- Basic Morphological Operations:
  - Closing = Dilation and then Erosion

    » Morphological closing is useful for filling small holes from an image while preserving the shape and size of the objects in the image.



Source:https://se.mathworks.com/help/images/morphological-dilation-and-erosion.html

Outline

- What is Image Processing?

- Color
- Linear Filtering
- Non-linear Filters / Thresholding
- Morphology
- <u>Connected Components Analysis</u>

- Summary

Connected Components Analysis

- Connencted Component Analysis checks each pixel of an image for connectivity with its neighboring pixels.
- Each group of connected pixels are considered as one component and are assigned the same label.

- Connectivity is established if two neighboring pixels share same or similar intensity/color value.
- The method works both binary. grayscale, or color images
- Different measures of connectivity arone possible (4-connectivity, or 8-connectivity are typical)

Summary

- We discussed about what Image Processing is.

- We learned about :
  - Color
    – RGB, other Color Spaces
  - Linear Filtering
    – convolution, cross-corelation
  - Non-linear Filters / Thresholding
  - Morphology
  - Connected Components Analysis

Lazaros Nalpantidis

Image Processing