Image Processing 03
Basic Tools in Image Processing
Part 1

SS 2020

Prof. Dr. Simone Frintrop

Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Outline

- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

The Image Processing Toolbox

Image processing tools: like tools for construction

A suitable tool for every task.

Sometimes, different tools can be used for the same task

Remember

Representation of Images

- The image A

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

- can also be written as intensity function f(x,y):

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

Element-wise vs Matrix Operations

Distinguish two types of operations on images:

- Element-wise operations: apply operation on pixel-by-pixel basis (more common)
- Matrix-operations: treat image as mathematical matrix and apply matrix operations known from matrix theory

- If not stated otherwise, we assume the element-wise product

Element-wise vs Matrix Operations

- Example: Given matrices A and B:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ and } \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

- The element-wise product (also Hadamard product) of A and B is:

$$\begin{bmatrix} \boxed{a_{11}} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \odot \begin{bmatrix} \boxed{b_{11}} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} \boxed{a_{11}b_{11}} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

Element-wise vs Matrix Operations

- Example: Given matrices A and B:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{and} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

- The matrix product of A and B is:

$$\begin{bmatrix} \boxed{a_{11} \quad a_{12}} \\ a_{21} \quad a_{22} \end{bmatrix} \begin{bmatrix} \boxed{\begin{matrix} b_{11} \\ b_{21} \end{matrix}} & \begin{matrix} b_{12} \\ b_{22} \end{matrix} \end{bmatrix} =$$

Element-wise vs Matrix Operations

- Example: Given matrices A and B:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ and } \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

- The matrix product of A and B is:

$$\begin{bmatrix} \boxed{a_{11} \quad a_{12}} \\ a_{21} \quad a_{22} \end{bmatrix} \begin{bmatrix} \boxed{\begin{matrix} b_{11} \\ b_{21} \end{matrix}} & \begin{matrix} b_{12} \\ b_{22} \end{matrix} \end{bmatrix} = \begin{bmatrix} \boxed{a_{11}b_{11} + a_{12}b_{21}} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Element-wise vs Matrix Operations

- Example: Given matrices A and B:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ and } \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

- The element-wise product (also Hadamard product) of A and B is:

$$\begin{bmatrix} \boxed{a_{11}} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \odot \begin{bmatrix} \boxed{b_{11}} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} \boxed{a_{11}b_{11}} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

- The matrix product of A and B is:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} \boxed{a_{11}b_{11} + a_{12}b_{21}} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Outline

- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Image Processing 03
Basic Tools in Image Processing
Part 2

SS 2020

Prof. Dr. Simone Frintrop

Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Outline

- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

Arithmetic Operations

- Arithmetic operations ($+ - \times \div$) between two images f and g are element-wise operations:

$$
\begin{aligned}
s(x,y) &= f(x,y) + g(x,y) \\
d(x,y) &= f(x,y) - g(x,y) \\
p(x,y) &= f(x,y) \times g(x,y) \\
v(x,y) &= f(x,y) \div g(x,y)
\end{aligned}
$$

- Variant: g is replaced by a constant (meaning: the constant is added to every pixel etc.)
- Arithmetic operations play an important role in image processing, with many applications

Arithmetic Operations: Addition

- Addition adds two images f and g element-wise:

$$s(x, y) = f(x, y) + g(x, y)$$

- A (not especially useful but illustrative) example:

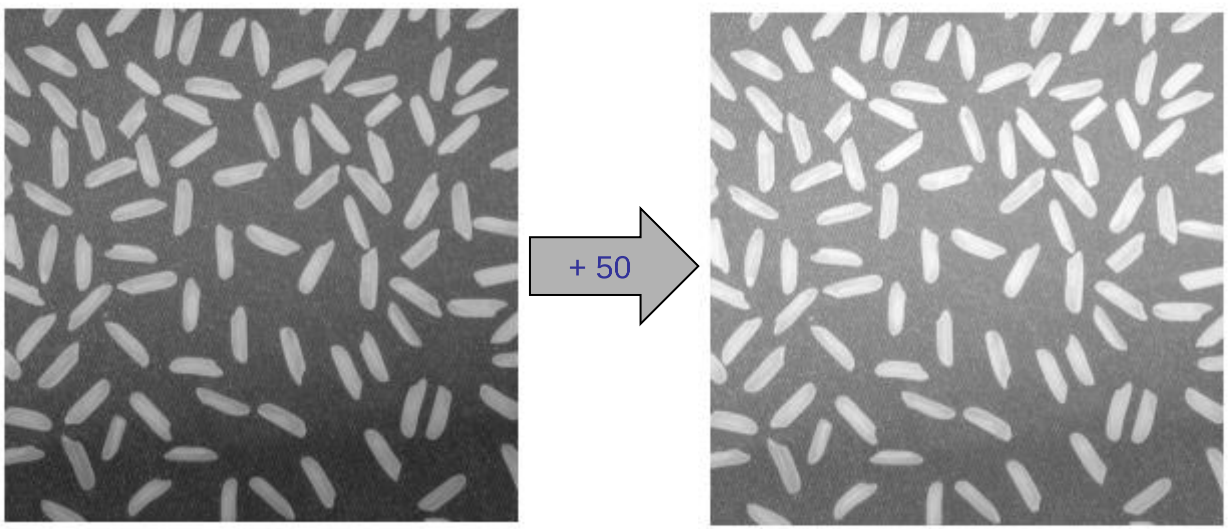
https://www.mathworks.com/help/images/ref/imadd.html

Arithmetic Operations: Addition

- Addition adds two images f and g element-wise:

$$s(x,y) = f(x,y) + g(x,y)$$

- g can also be a constant.
- What is the effect of adding a constant to f ?
- The image will be brighter:



https://www.mathworks.com/help/images/ref/imadd.html

Arithmetic Operations: Addition

- Addition adds two images f and g element-wise:

$$s(x,y) = f(x,y) + g(x,y)$$

- Which problems can occur here?
- Overflow (values exceed maximal pixel value)
- Solution?
  - Set values to the allowed maximum (saturation)
  - Wrap around (start by 0 again)
  - Scale values of result image to span allowed range [0..K]:

$$g_m = g - min(g)$$
$$g_s = K[g_m / max(g_m)]$$

(these are elementwise subtraction and division

Arithmetic Operations: Addition

Scaling image to span range [0..K]:

$$g_m = g - min(g)$$
$$g_s = K[g_m / max(g_m)]$$

Calculate the new pixel values for the image [10 50 110]

Arithmetic Operations: Addition

Averaging for reducing noise:

- Suppose a corrupted image g is formed by the addition of noise η to the original image signal f:

$$g(x, y) = f(x, y) + \eta(x, y)$$

- If we would know the noise, we could simply subtract it from f
- Since we do not know the noise, we can use the law of large numbers to reduce noise:

Averaging for Noise Reduction

Law of large numbers (Gesetz der großen Zahlen):

- Let X be a random variable with expected value $\mathbb{E}(X)$. As the number of trials (drawing X) increases, their average approaches their theoretical mean.
  (Sei X eine Zufallsvariable mit Erwartungswert $\mathbb{E}(X)$. Je häufiger das Zufallsexperiment durchgeführt wird, desto mehr nährt sich der Durchschnitt von X dem Erwartungswert an.)

Beispiel: Wurf einer Münze

Anzahl Würfe	davon Kopf		Verhältnis		absoluter Abstand	relativer Abstand
	theoretisch	beobachtet	theoretisch	beobachtet		
100	50	48	0.500	0.480	2	0.02
1000	500	491	0.500	0.491	9	0.009
10000	5000	4970	0.500	0.497	30	0.003

https://mathepedia.de/Gesetz_der_groszen_Zahlen.html

Averaging for Noise Reduction

Remember:

- A corrupted image g is formed by the addition of noise η to the original image signal f:

$$g(x,y) = f(x,y) + \eta(x,y)$$

- The law of large numbers tells us: if we average many corrupted images, the average will approach the original signal f

Averaging for Noise Reduction

Noise reduction method:

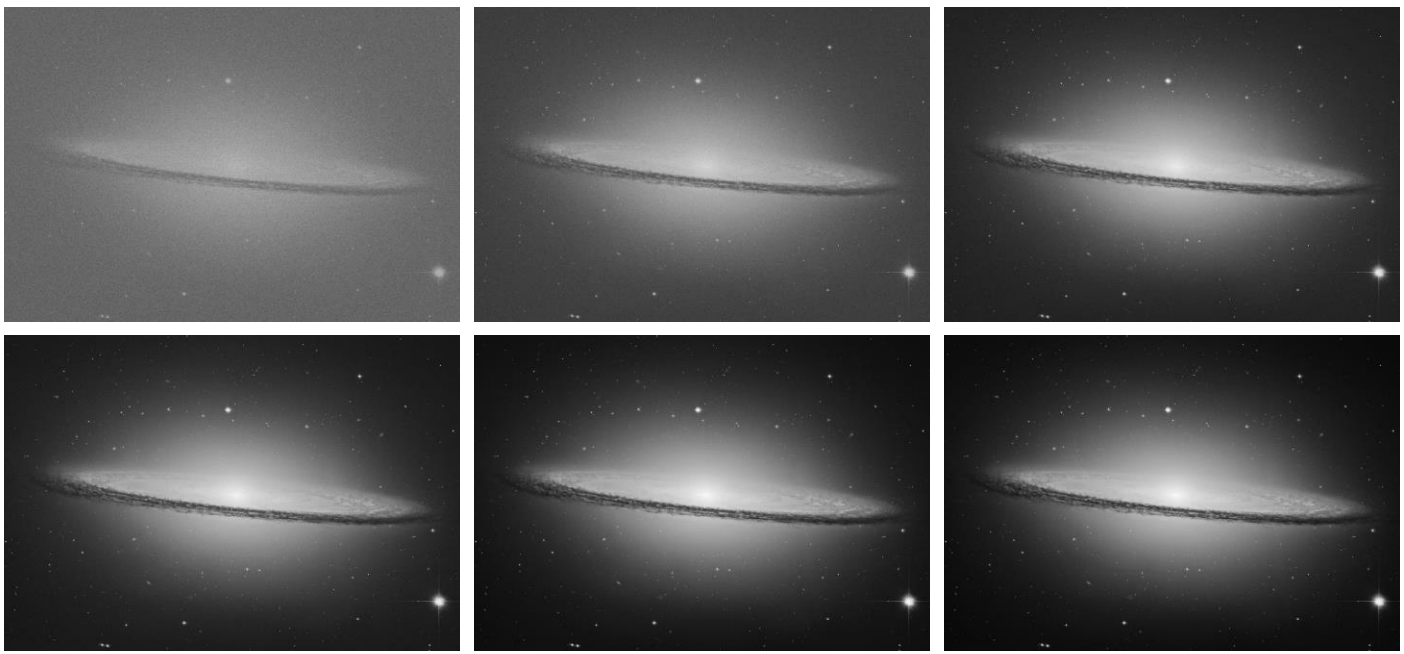
- Capture K different (noisy) images g_i
- Average all K noisy images to obtain image $\bar{g}$ with reduced noise:

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

- The expected value of the average image $\bar{g}$ is the original signal f : $E\{\bar{g}(x,y)\} = f(x,y)$
- The variance (and standard deviation) of $\bar{g}$ decreases as K increases (see Gonzales/Woods, example 2.5)

Averaging for Noise Reduction

- Application: astronomy, imaging under low light levels causes noisy images
- Averaging of K images results in:



a b c
d e f

FIGURE 2.29 (a) Sample noisy image of the Sombrero Galaxy. (b)-(f) Result of averaging 10, 50, 100, 500, and 1,000 noisy images, respectively. All images are of size 1548×2238 pixels, and all were scaled so that their intensities would span the full [0, 255] intensity scale. (Discovered in 1767, the Sombrero Galaxy is 28 light years from Earth. Original image courtesy of NASA.)

[Gonzales/Woods]

Arithmetic Operations: Addition

Blending:

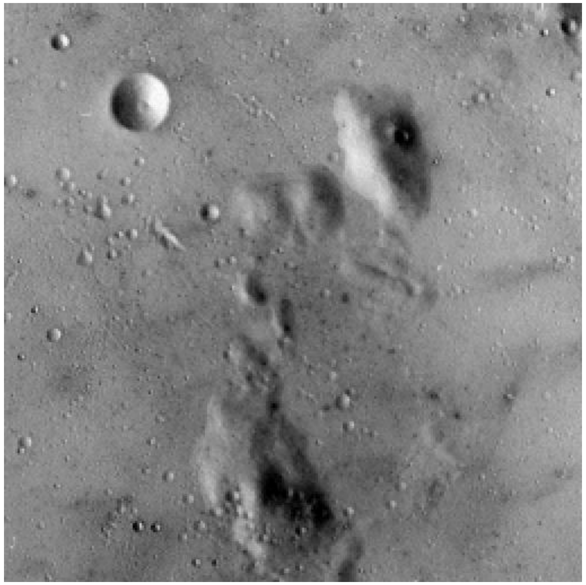
- Blending is weighted addition of two images with a weight factor α:

$$g(x) = (1-\alpha) f_0(x) + \alpha f_1(x)$$

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_core/py_image_arithmetics/py_image_arithmetics.html

Arithmetic Operations: Addition

Another example of Blending: (with α = 0.5)


+

=


Outline

- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Image Processing 03
Basic Tools in Image Processing
Part 3

SS 2020

Prof. Dr. Simone Frintrop

Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Outline

- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

Arithmetic Operations: Subtraction

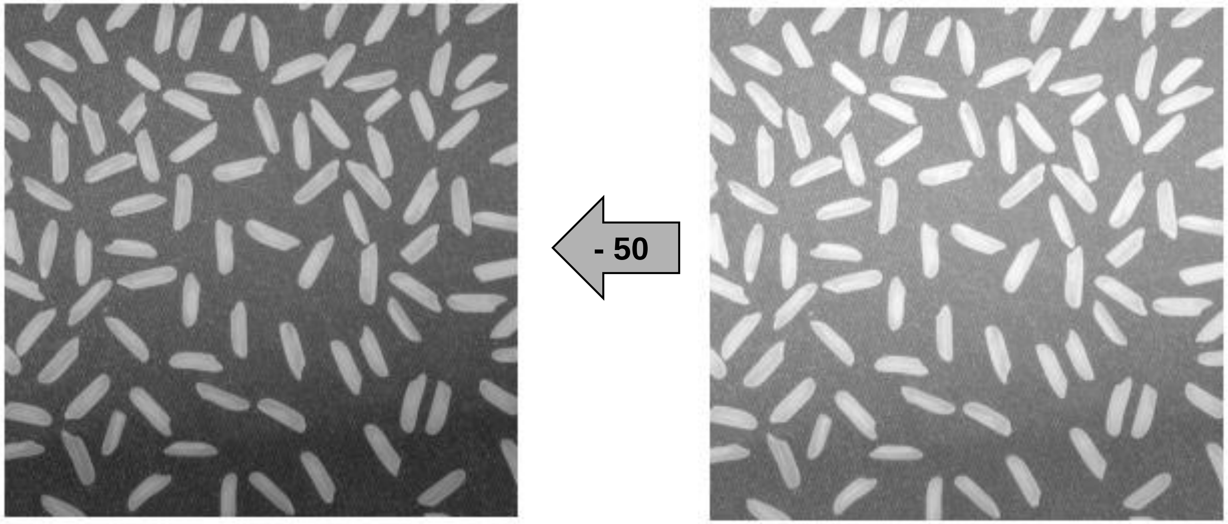
- Subtraction subtracts the values of image g from image f element-wise:

$$d(x,y) = f(x,y) - g(x,y)$$

- Applications:
  - Enhance/visualize differences between images
  - Background subtraction
  - Mask parts of the image

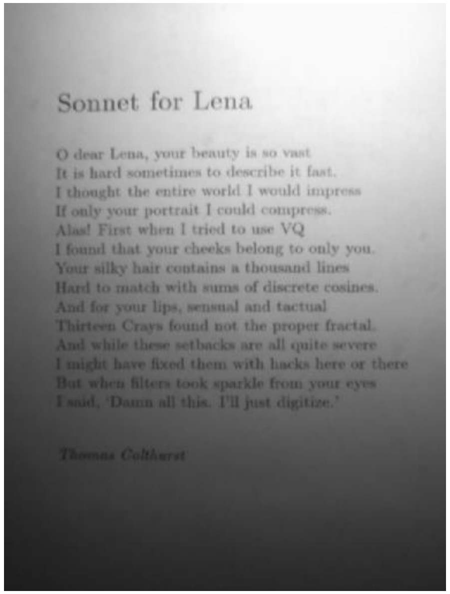
Arithmetic Operations: Subtraction

- Subtraction to make image darker
  (but note that there are better methods for darkening images, like scaling…)



https://www.mathworks.com/help/images/ref/imadd.html

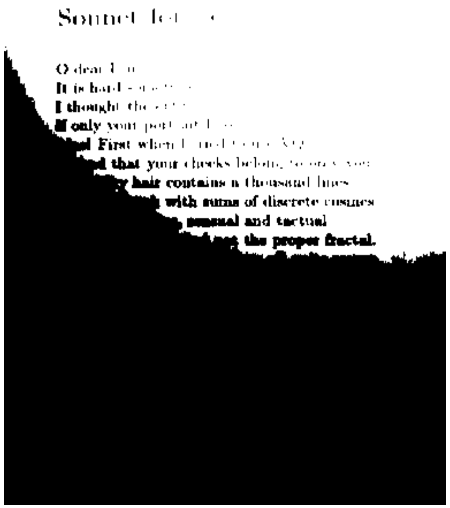
Arithmetic Operations: Subtraction


Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

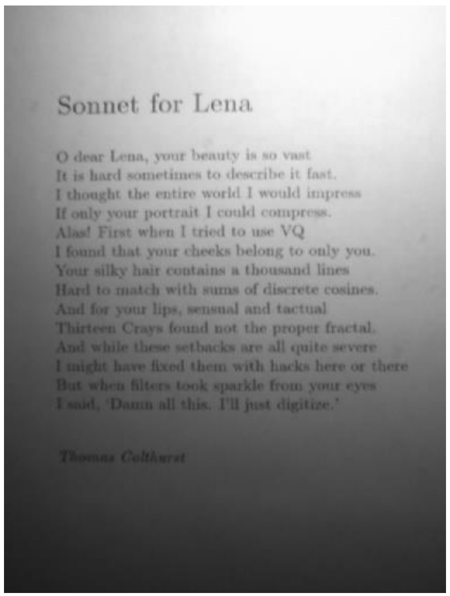
Original image under bad illumination conditions



Result from simple thresholding

https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixsub.htm

Arithmetic Operations: Subtraction



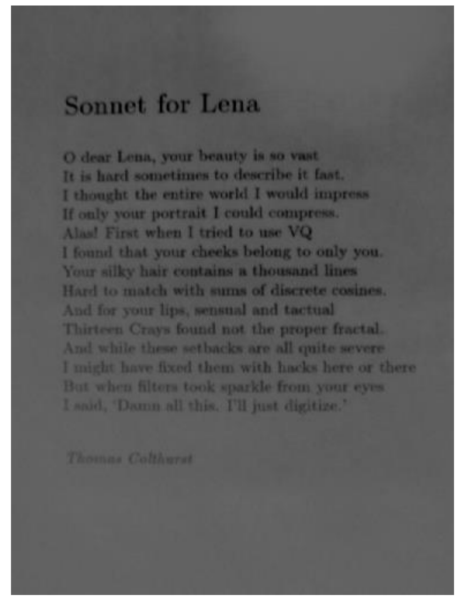
Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

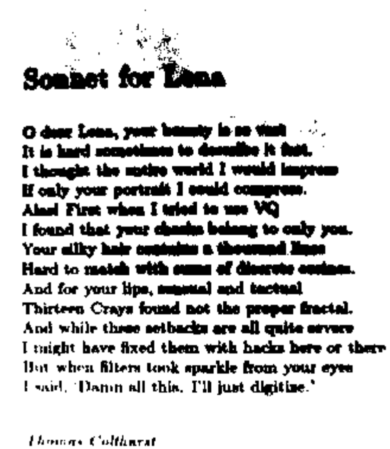
Thomas Colthurst

Original image under bad illumination conditions

„lightfield image“:
a white paper captured under same illumination conditions



Subtract the lightfield image from the original image

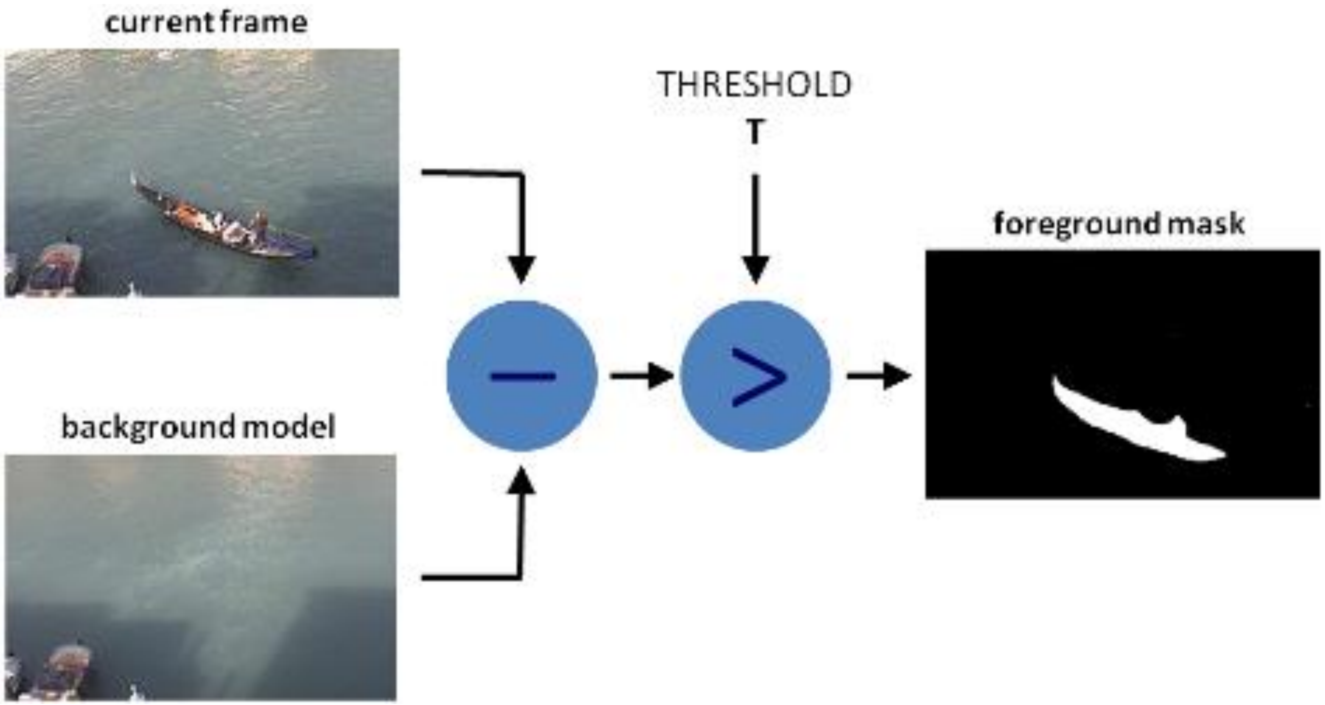


et voilà: better result than previously

https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixsub.htm

Arithmetic Operations: Subtraction

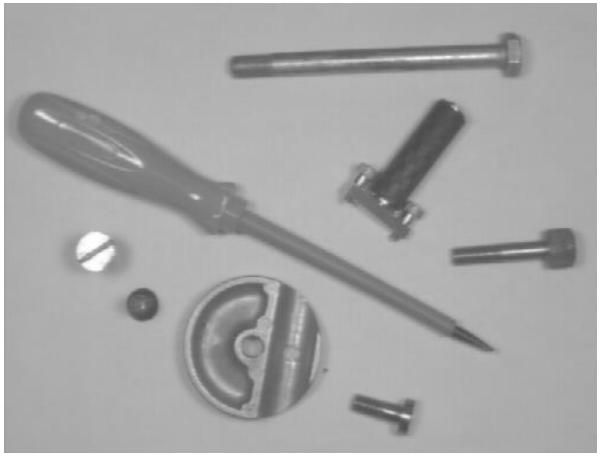
Background subtraction to extract foreground image:



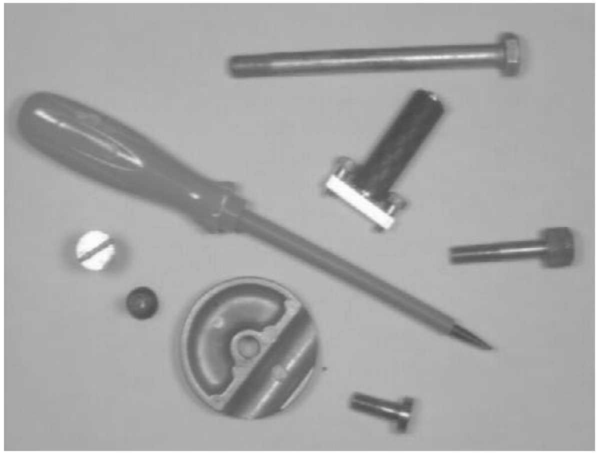
https://docs.opencv.org/3.4/d1/dc5/tutorial_background_subtraction.html

Arithmetic Operations: Subtraction

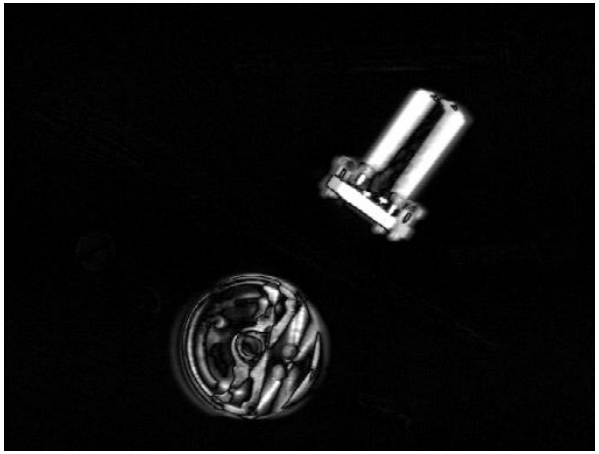
Background subtraction for change detection:



Original image



Similar image with minor differences



Subtracting the two images visualizes the changes

https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixsub.htm

Arithmetic Operations: Subtraction

Subtraction to detect motion:

http://tutorial.simplecv.org/en/latest/examples/image-math.html

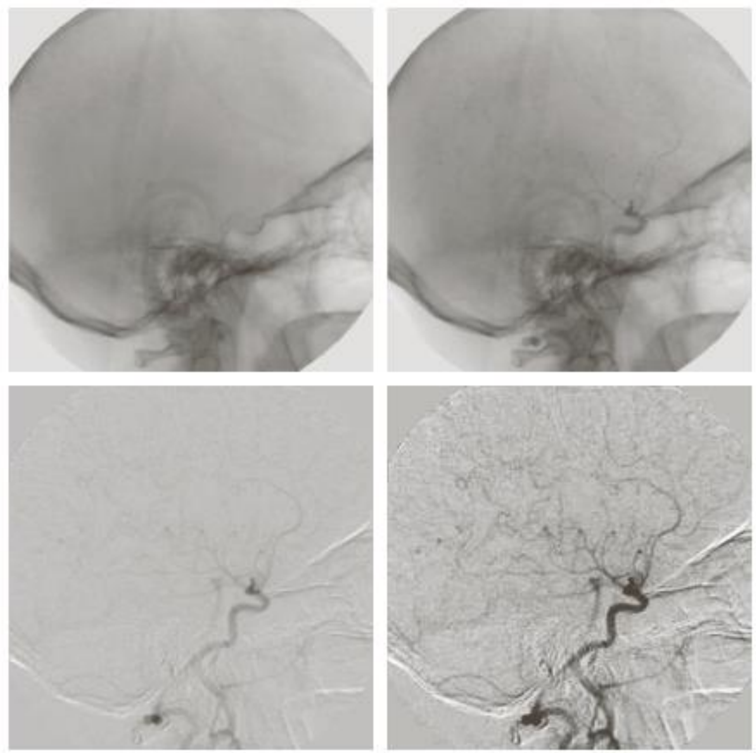
Arithmetic Operations: Subtraction

Subtraction for medical imaging: mask mode radiography:

$$g(x, y) = f(x, y) - h(x, y)$$

a b
c d

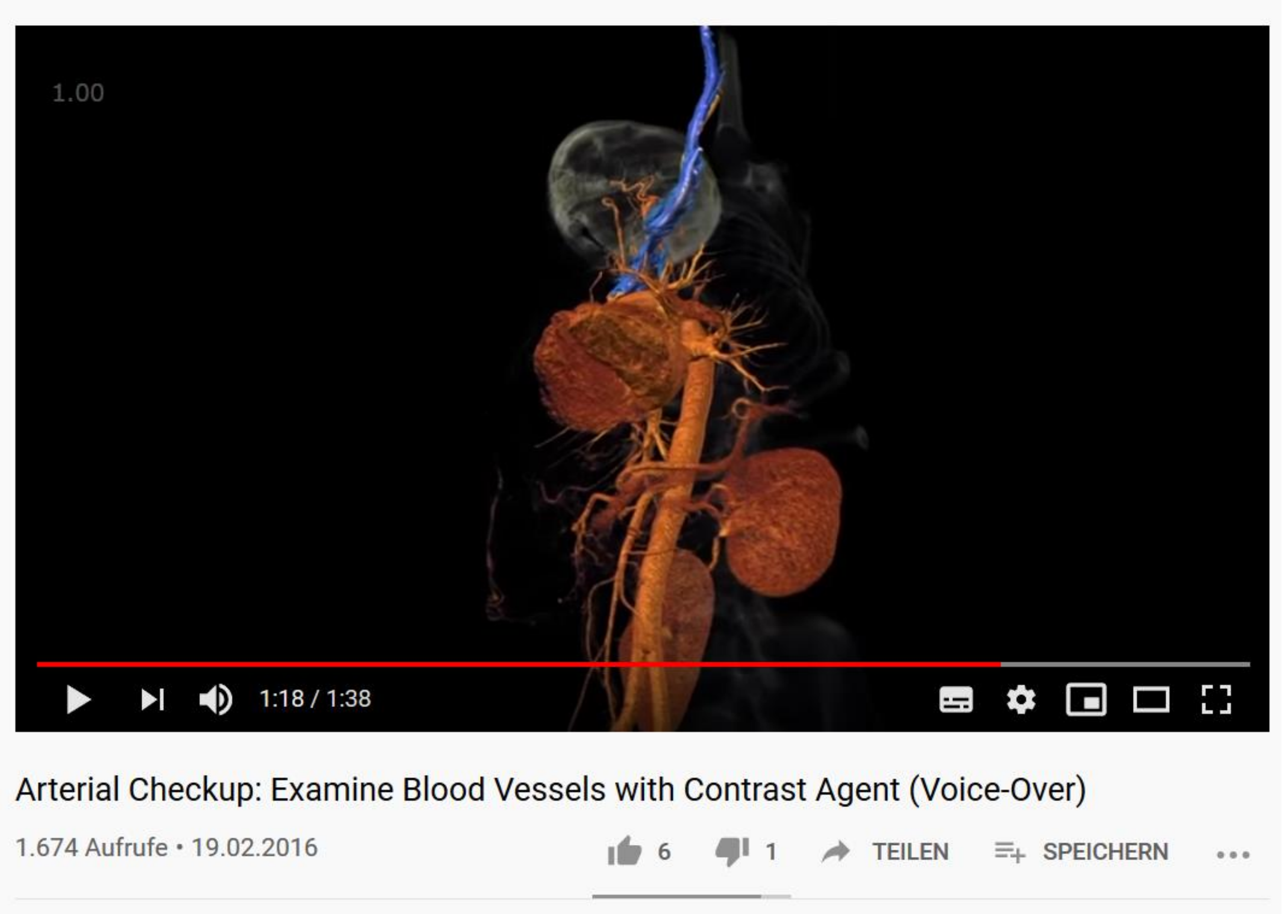
FIGURE 2.32 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of the Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)



[Gonzales/Woods]

Arithmetic Operations: Subtraction

Subtraction for medical imaging: mask mode radiography:



https://www.youtube.com/watch?v=b-u4K5wtWSQ (Fraunhofer) (Injecting contrast medium)

Outline

- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

Image Processing 03
Basic Tools in Image Processing
Part 4

SS 2020

Prof. Dr. Simone Frintrop

Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Outline

- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

Arithmetic Operations: Multiplication

- Multiplication multiplies the pixels of two images f and g element-wise:

$$p(x, y) = f(x, y) \times g(x, y)$$

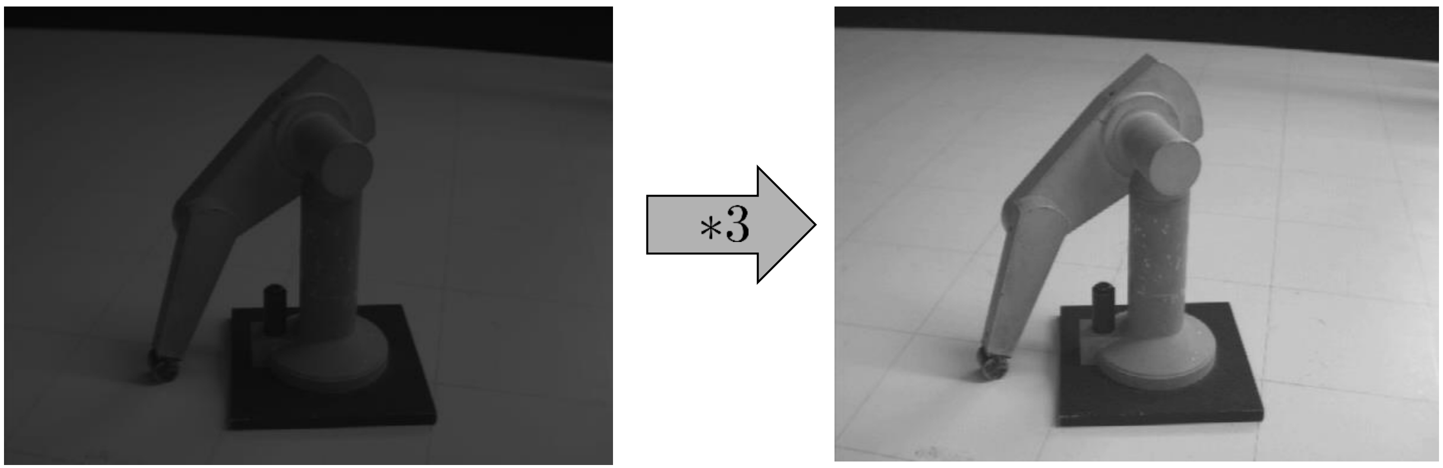
- If g is replaced by a constant c, this is called ***scaling:***

$$p(x, y) = f(x, y) \times c$$

- What is the effect of scaling with a factor > 1?
- A: brighter image
- And a factor < 1 ?
- A: darker image
- Scaling produces more natural effects than addition and subtraction since it preserves the relative contrast

Arithmetic Operations: Multiplication

- Example of ***scaling***:

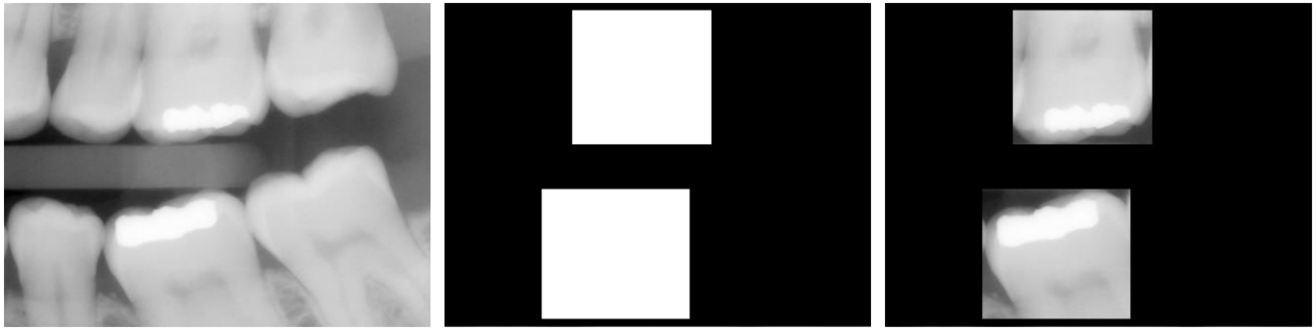


- As usual, be aware not to exceed the allowed intensity range (or transform afterwards to allowed range)

https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixmult.htm

Arithmetic Operations: Multiplication

- Multiplication for masking:
- Use a binary image to determine the parts which should be visible (multiply regions to be preserved with 1, others with 0)



a b c
FIGURE 2.34 (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

- Alternative: use AND operator (see later)

[Gonzales/Woods]

Arithmetic Operations: Division

- Division divides the pixels of image f by the values from g element-wise:

$$v(x, y) = f(x, y) \div g(x, y)$$

- Example: we can model an image g captured by an imperfect sensor by the product of the original signal f and a shading function h:

$$g(x, y) = f(x, y) * h(x, y)$$

- After re-arranging, this becomes:

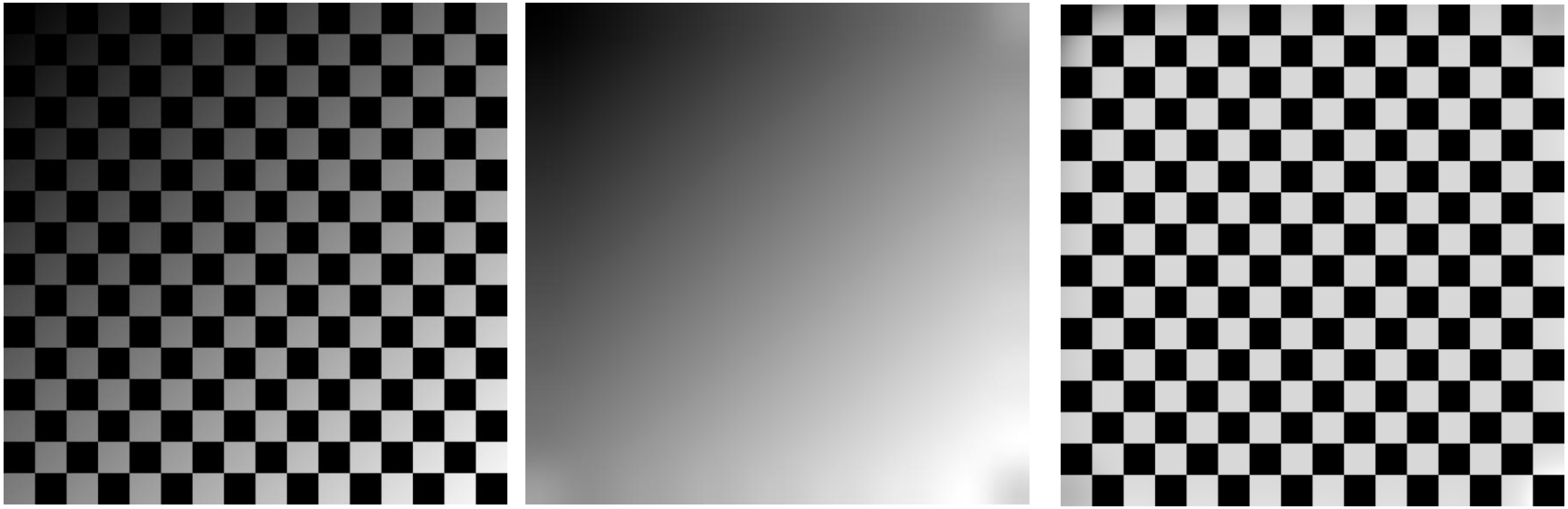
$$\mathrm{f(x,y)=g(x,y)/h(x,y)}$$

- Thus, we can use division to estimate the original signal

Arithmetic Operations: Division

Division for shading correction:

- Obtain the shading image h by capturing a white surface
- Divide each element of g by each element of h :



Shaded input g | Shading pattern h | Corrected image f

a b c

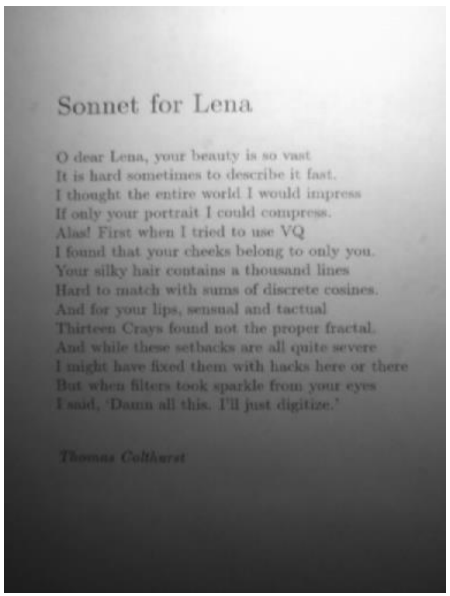
FIGURE 2.33 Shading correction. (a) Shaded test pattern. (b) Estimated shading pattern. (c) Product of (a) by the reciprocal of (b). (See Section 3.5 for a discussion of how (b) was estimated.)

[Gonzales/Woods]

Arithmetic Operations: Division

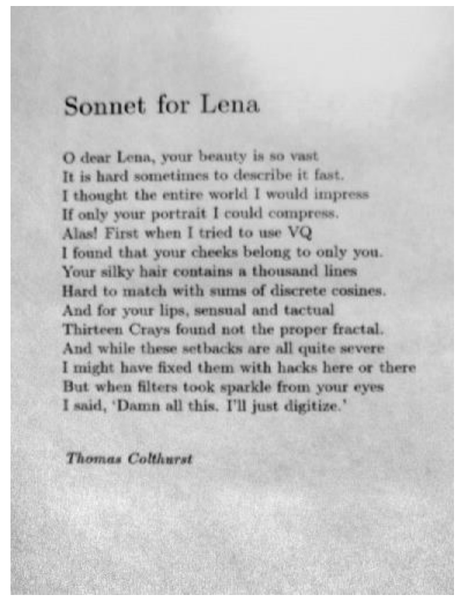
Division for shading correction:

- Here again the example from earlier, now using division:

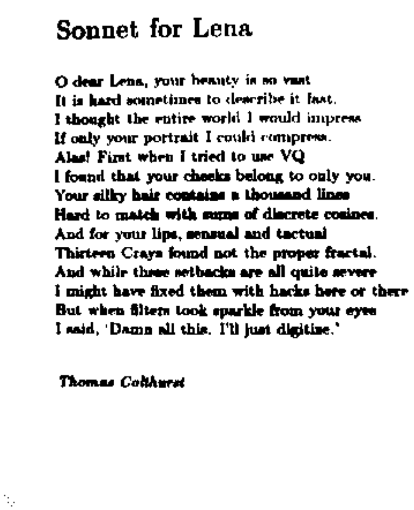


Original image under bad illumination conditions

„lightfield image“: a white paper captured under same illumination conditions



Divide the lightfield image from the original image

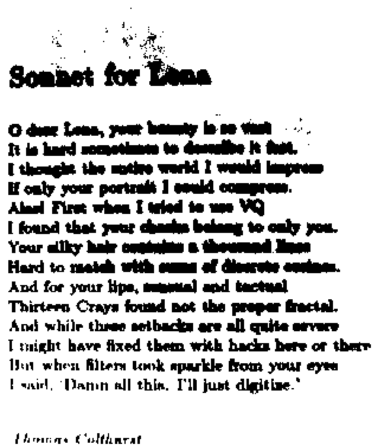


et voilà: better result than previously

https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixdiv.htm

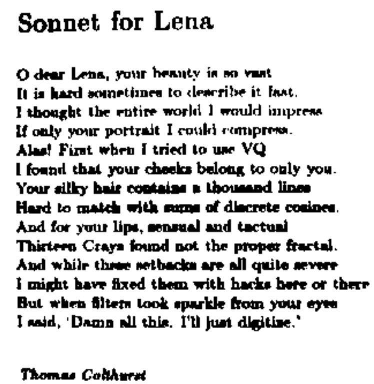
Shading Correction

Subtraction vs Division:


Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst


Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Subtraction

Division

Division usually obtains better results

https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixdiv.htm

Outline

- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Image Processing 03
Basic Tools in Image Processing
Part 5

SS 2020

Prof. Dr. Simone Frintrop

Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Outline

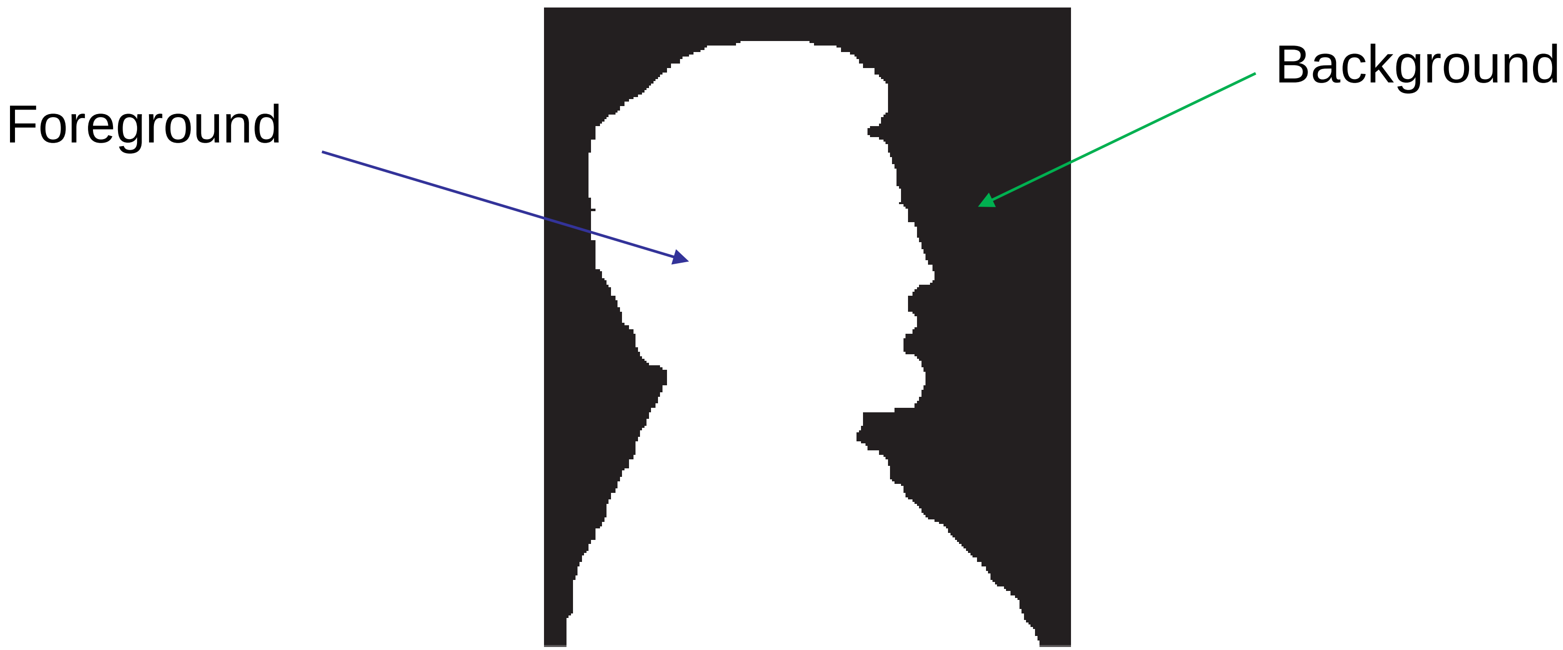
- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

Binary Image Processing

- Binary images: 1 bit per pixel, only two colors, usually black and white
- Two types of operations for binary images:
  - Set operations
  - Logical operations

Set Operations

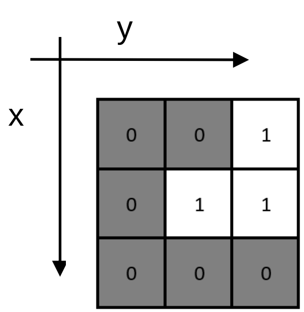
- Set-based image processing: images as sets of pixels
- Consider binary image as consisting of foreground (object) and background



Set Operations

- Set-based image processing: images as sets of pixels
- Consider binary image as consisting of foreground (object) and background
- We model only foreground (everything else is background) as a set of pixels: set of (x, y) coordinates

- Example:



A binary image

and the corresponding set:

$$I = \{(0,2), (1,1), (1,2)\}$$

Set Operations

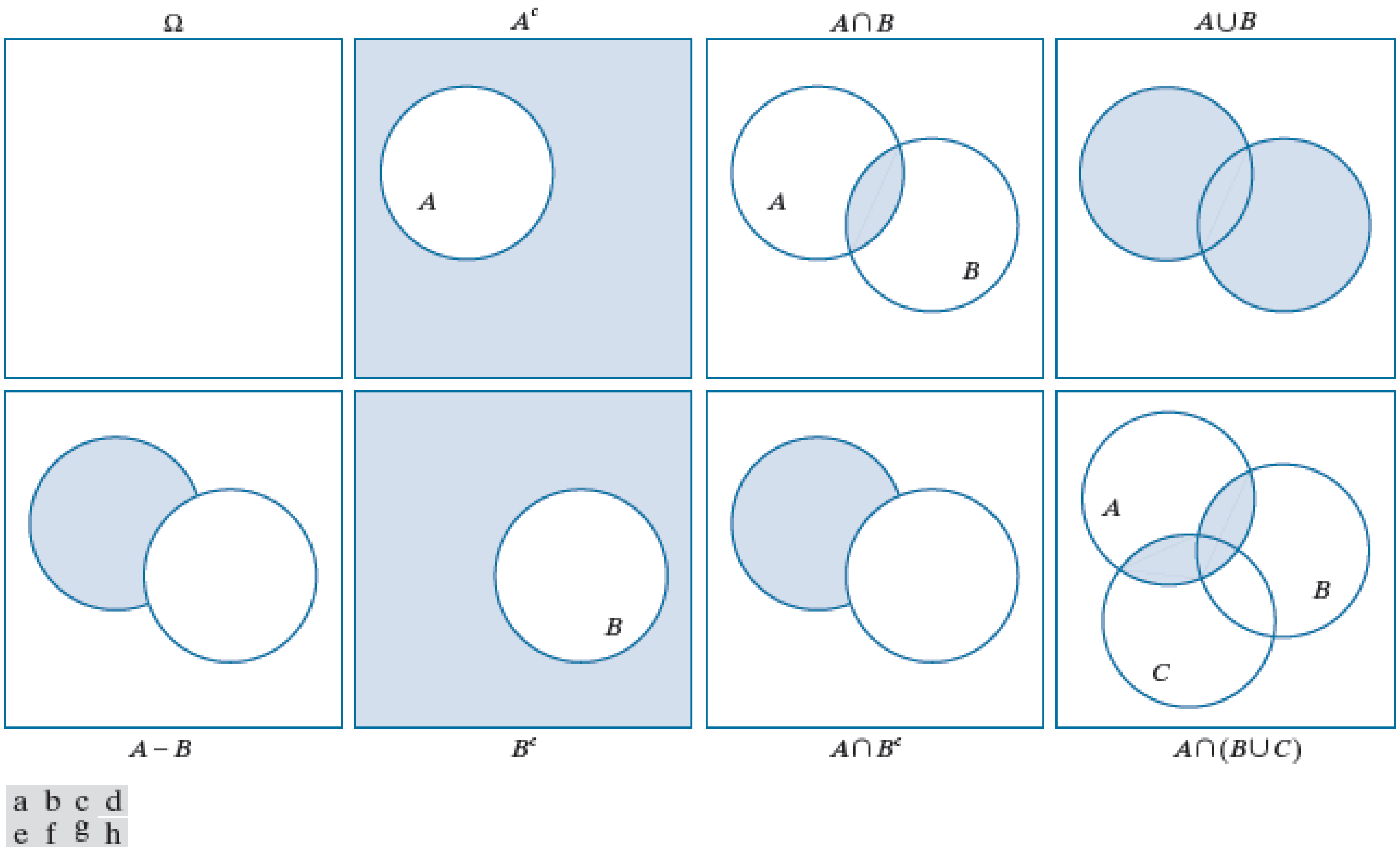


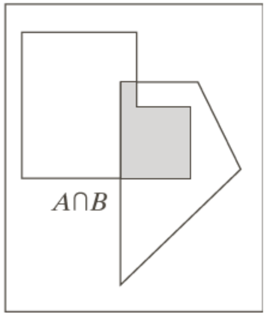
FIGURE 2.35 Venn diagrams corresponding to some of the set operations in Table 2.1. The results of the operations, such as A^c, are shown shaded. Figures (e) and (g) are the same, proving via Venn diagrams that $A - B = A \cap B^c$ [see Eq. (2-40)].

Set Operations

Example application:

Do objects A and B overlap?

Determine the intersection $A \cap B$. If not empty, they overlap.



[Gonzalez/Woods]

Set Operations

Set theory is also used in morphological image processing to improve binary images (remove noise, fill holes, etc)


Original image


Eroded image

More later in lecture “Morphological image processing”

[Gonzalez/Woods]

Set Operations

- Set operations can be modified to operate also on grayscale images:

Original image Complement

- We will not cover this here. Details in Gonzales Woods.

[Gonzalez/Woods]

Outline

- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Image Processing 03
Basic Tools in Image Processing
Part 6

SS 2020

Prof. Dr. Simone Frintrop

Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Outline

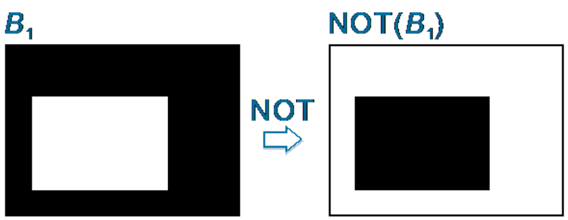
- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

Logical Operations

- Logical operations on images operate on binary images
- Pixel values are interpreted as truth values (0: FALSE, 1: TRUE)
- Apply a logical operation: apply the rules from a truth table element-wise to input image(s)

TABLE 2.2 Truth table defining the logical operators AND($\wedge$), OR($\vee$), and NOT($\sim$).

a	b	a AND b	a OR b	NOT(a)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0



[Gonzales/Woods]

Logical Operations

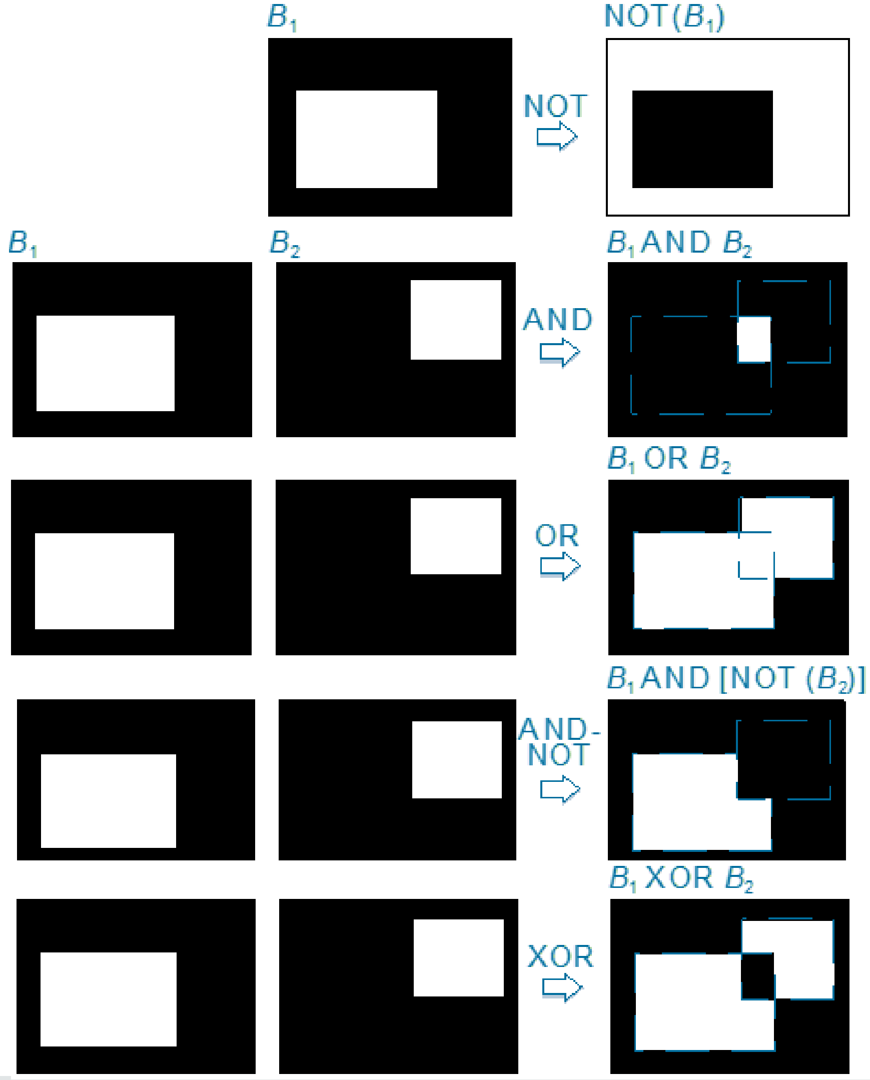
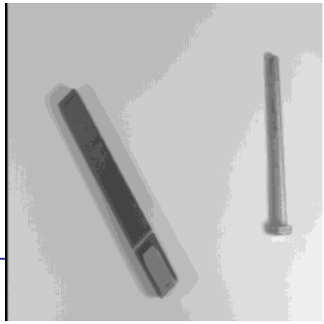
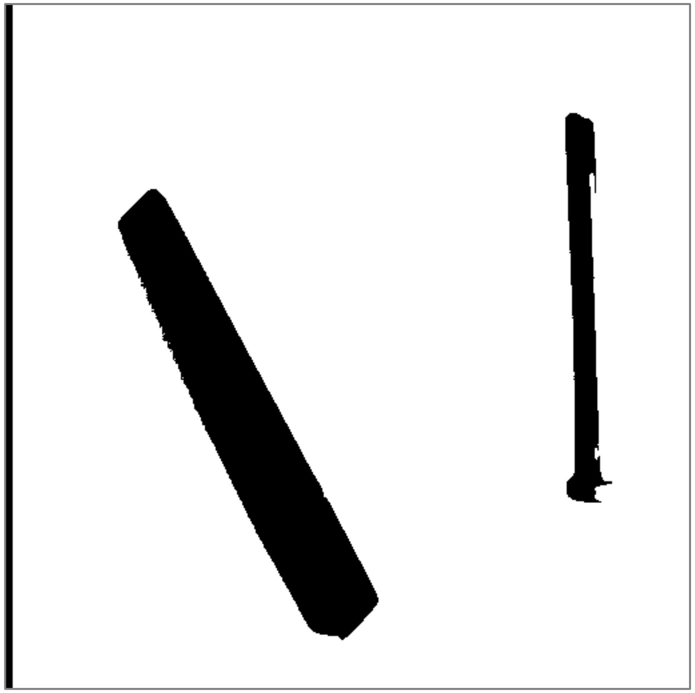


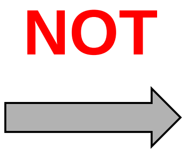
FIGURE 2.37 Illustration of logical operations involving foreground (white) pixels. Black represents binary 0's and white binary 1's. The dashed lines are shown for reference only. They are not part of the result.

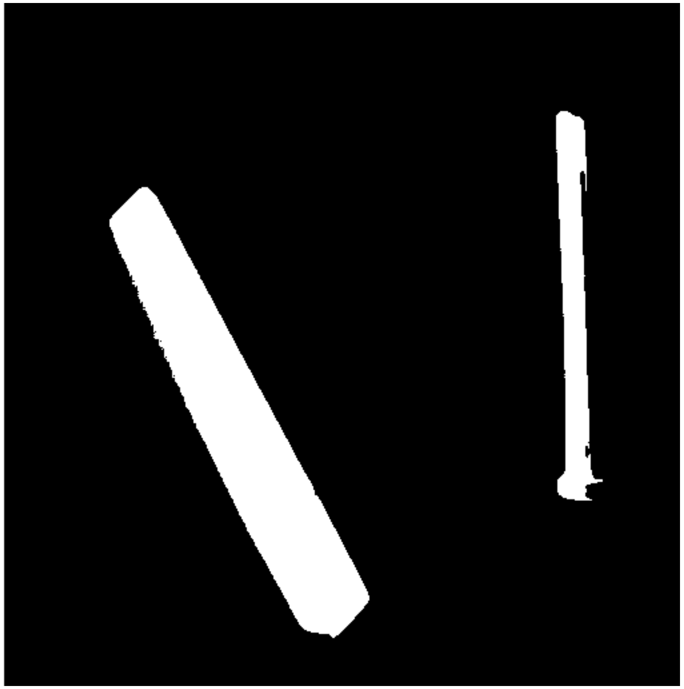
[Gonzales/Woods]

Logical Operations





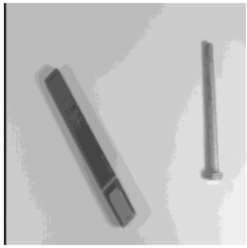


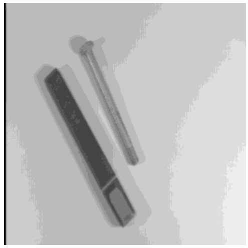


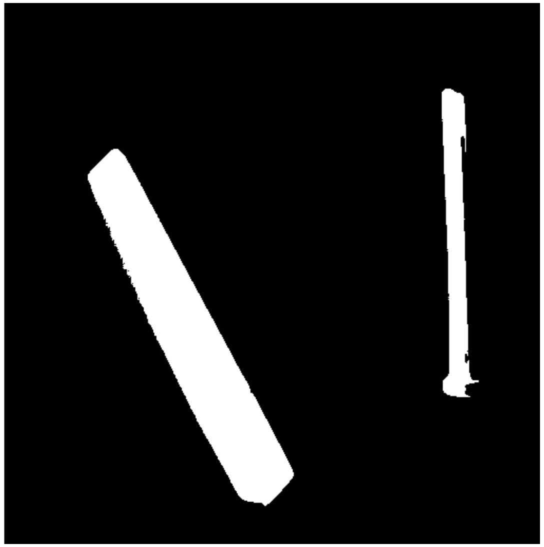
- The NOT operator is sometimes also called “invert”
- If thresholding shows the objects (foreground) in dark instead of white, we need NOT

[https://homepages.inf.ed.ac.uk/rbf/HIPR2/invert.htm]

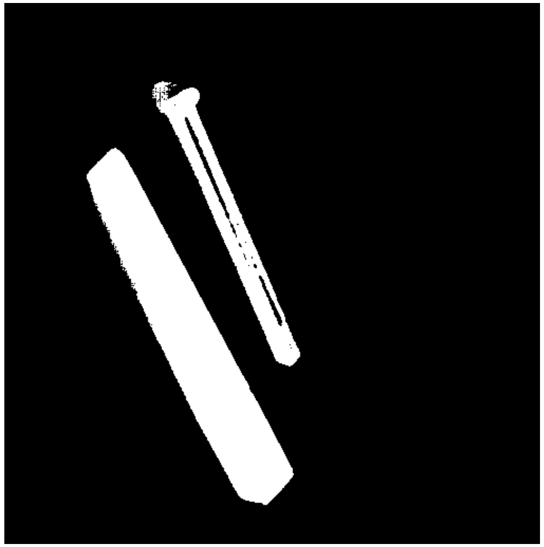
Logical Operations



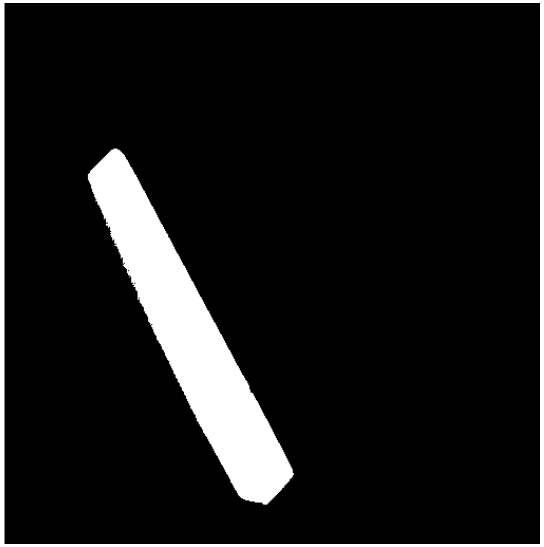




AND

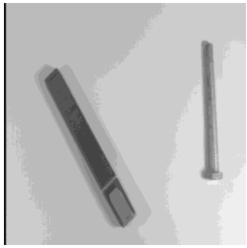
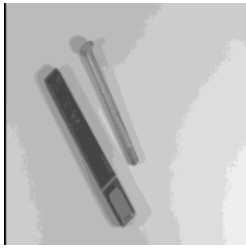


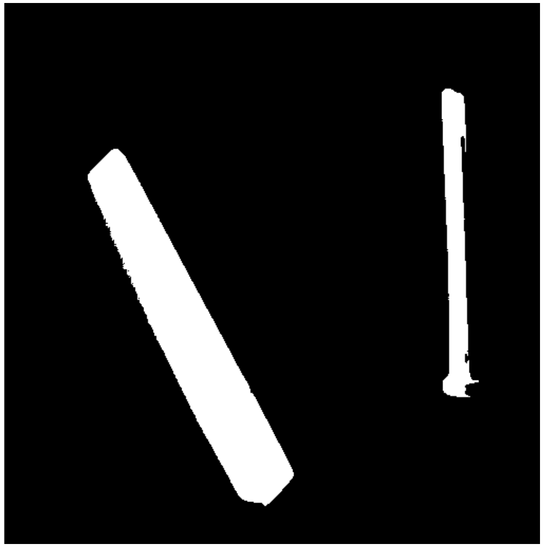
=



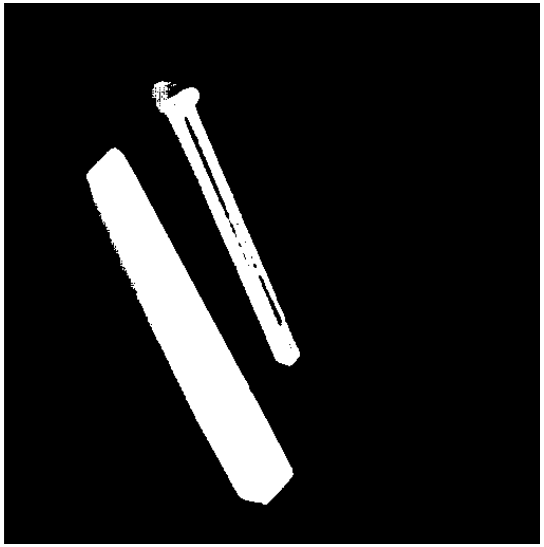
[https://homepages.inf.ed.ac.uk/rbf/HIPR2/or.htm]

Logical Operations

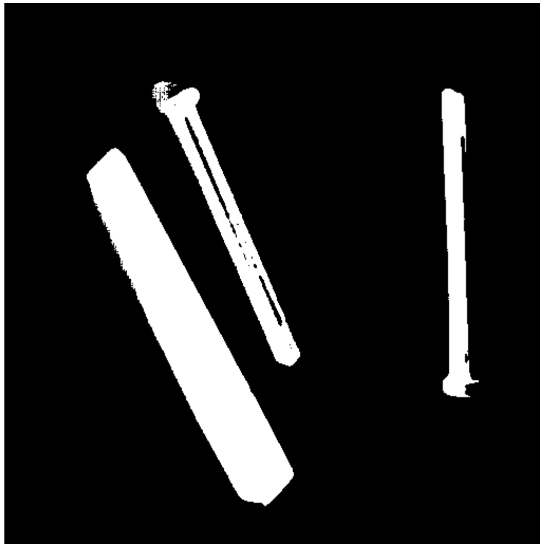





OR

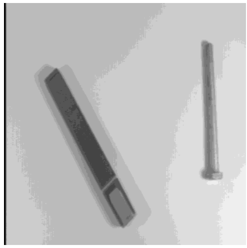


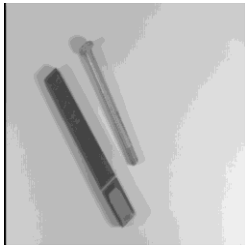
=

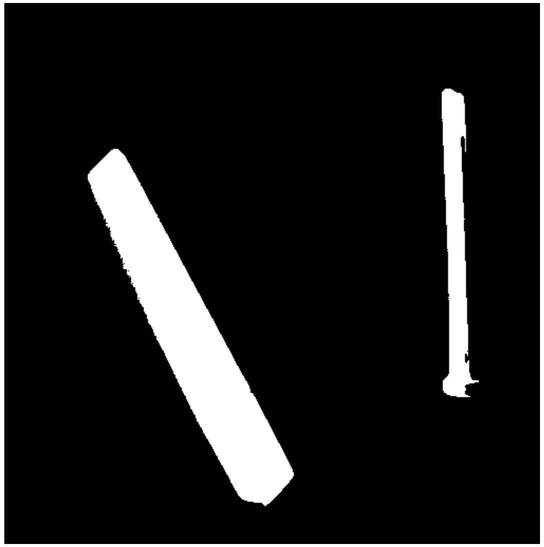


[https://homepages.inf.ed.ac.uk/rbf/HIPR2/or.htm]

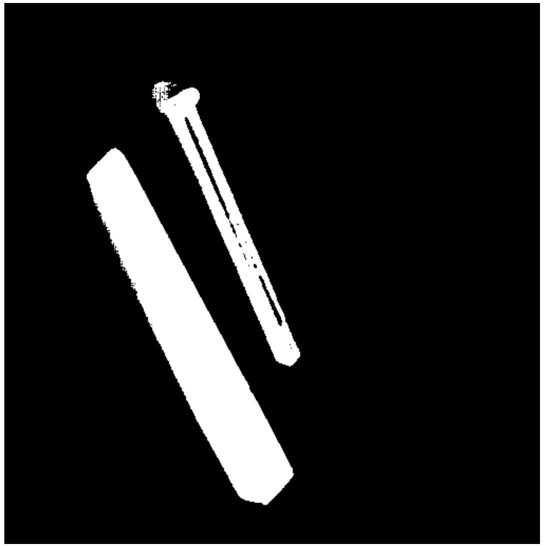
Logical Operations



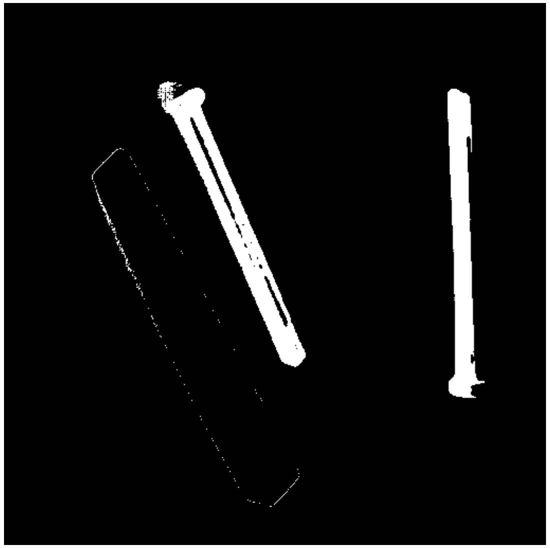




XOR

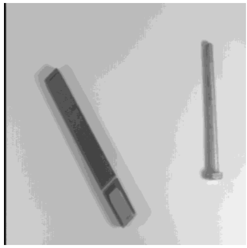


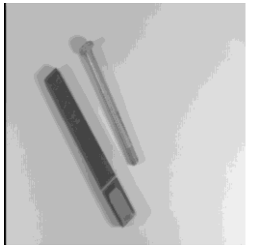
=



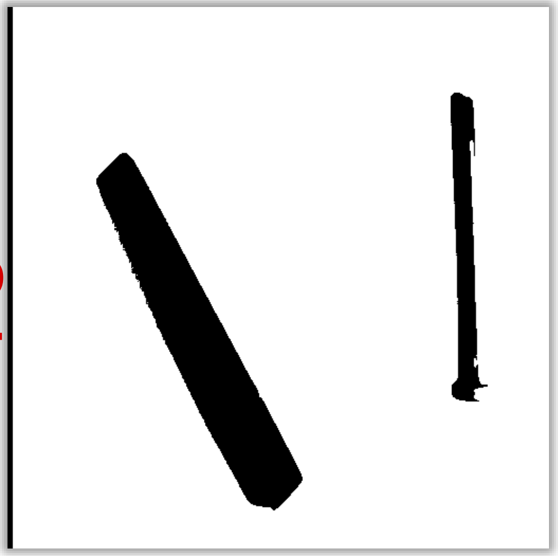
[https://homepages.inf.ed.ac.uk/rbf/HIPR2/xor.htm]

Logical Operations

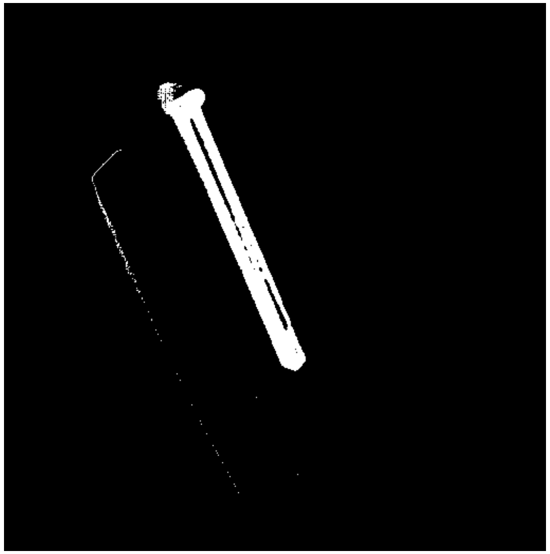




AND NOT

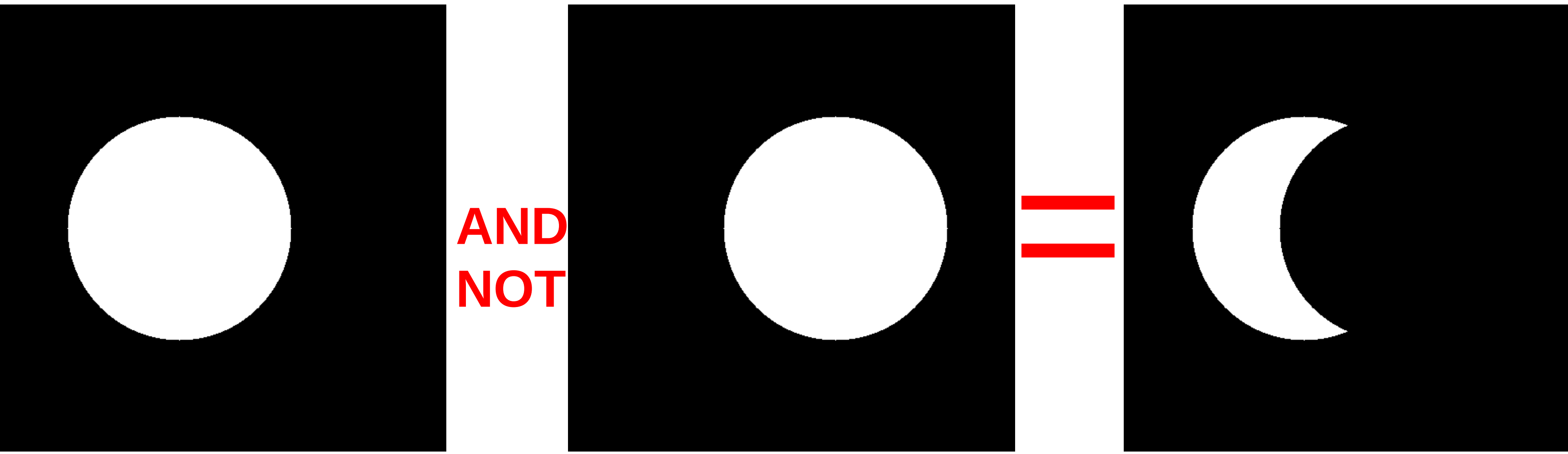


=



[https://homepages.inf.ed.ac.uk/rbf/HIPR2/xor.htm]

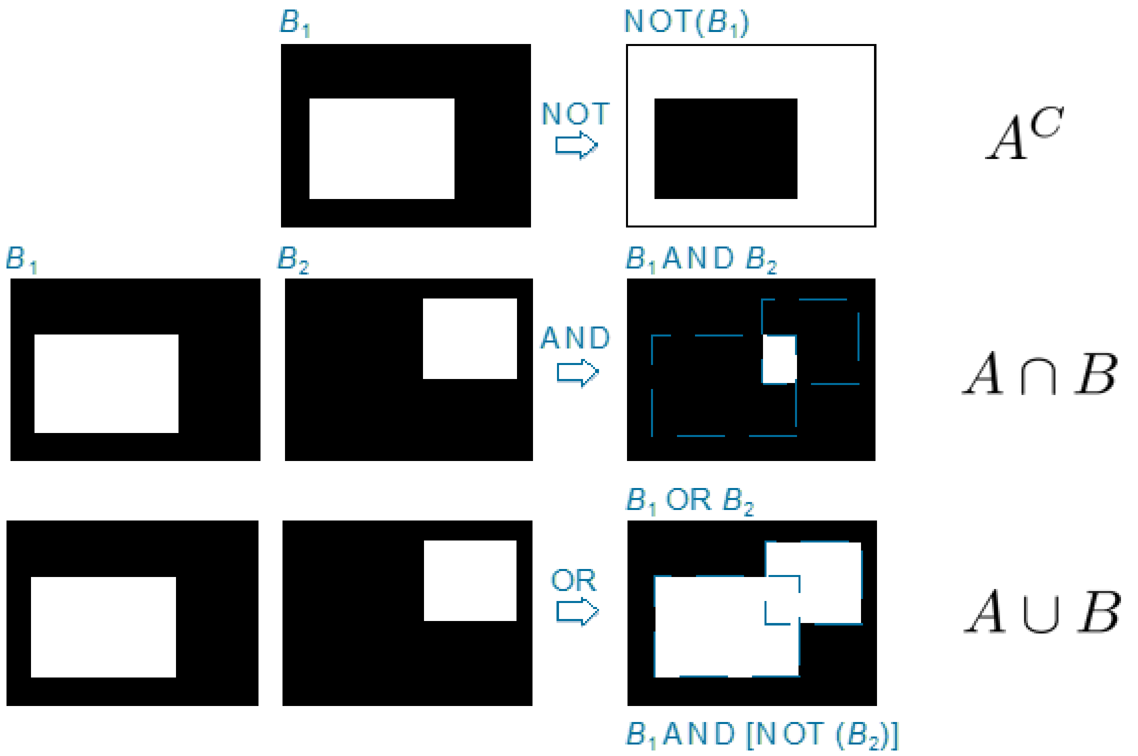
Logical Operations



[https://homepages.inf.ed.ac.uk/rbf/HIPR2/xor.htm]

Logical vs Set Operations

- Each logical operation can be replaced by a set operation and vice versa:



[Gonzalez/Woods]

Outline

- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

Image Processing 03
Basic Tools in Image Processing
Part 7

SS 2020

Prof. Dr. Simone Frintrop

Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Outline

- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations

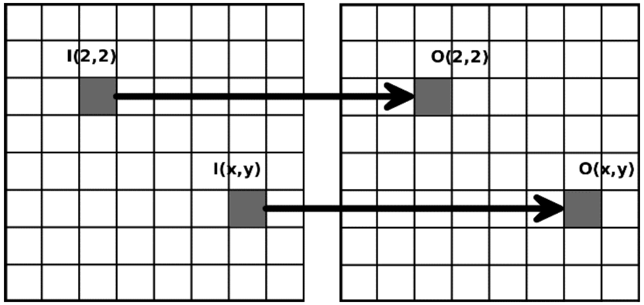
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Spatial Operations

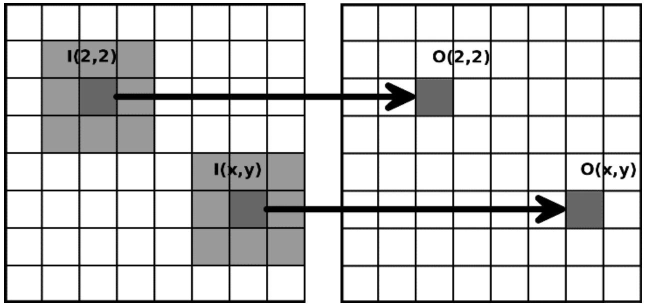
Spatial operations are operations directly applied to the pixels of an image.

We distinguish 4 categories:

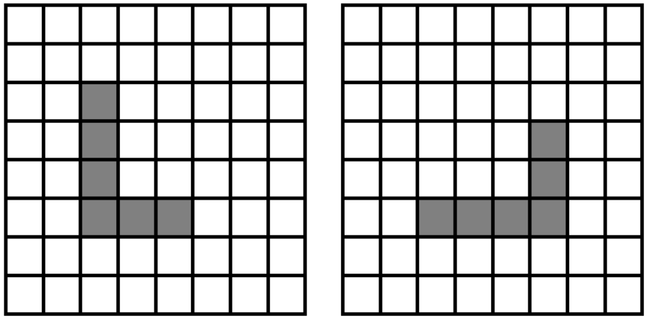
Point operations



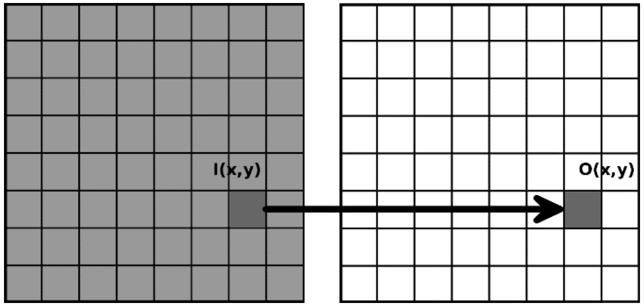
Neighborhood operations:



Geometric transformations



Global operations:

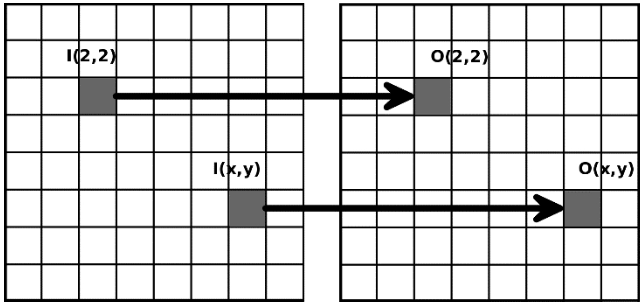


Spatial Operations

Spatial operations are operations directly applied to the pixels of an image.

We distinguish 4 categories:

Point operations

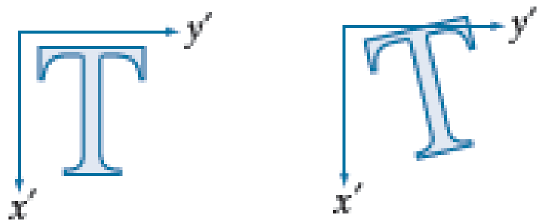


Transform image by adjusting single pixels
E.g.: contrast adjustment, color transformation
More in lecture: transformations

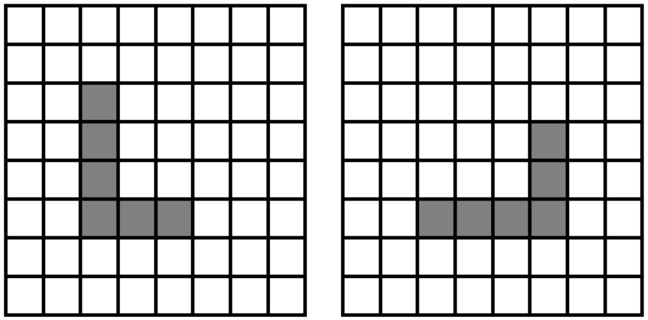
Spatial Operations

Spatial operations are operations directly applied to the pixels of an image.

We distinguish 4 categories:



Geometric transformations



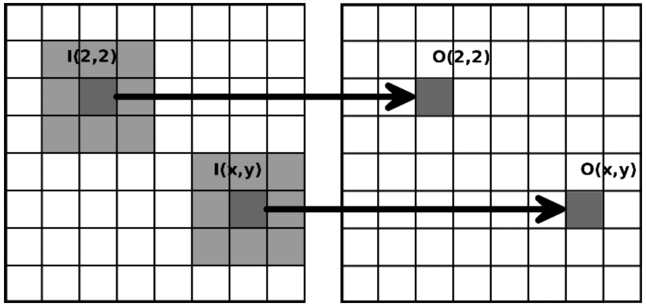
change a pixel value according to a geometric transformation
E.g.: scaling or rotation
More in lecture on transformations

Spatial Operations

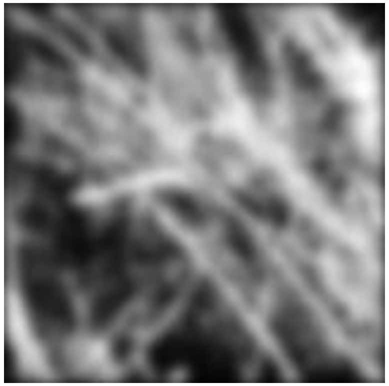
Spatial operations are operations directly applied to the pixels of an image.

We distinguish 4 categories:

Neighborhood operations:



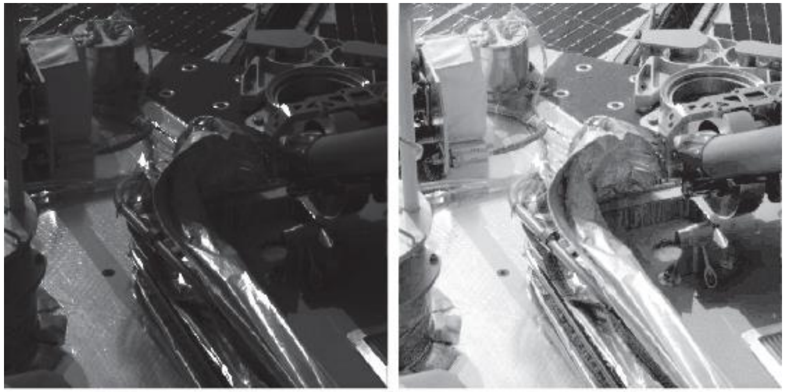
change a pixel value according to a local neighborhood
E.g.: smoothing or edge detection
More in lecture on Spatial Filters



Spatial Operations

Spatial operations are operations directly applied to the pixels of an image.

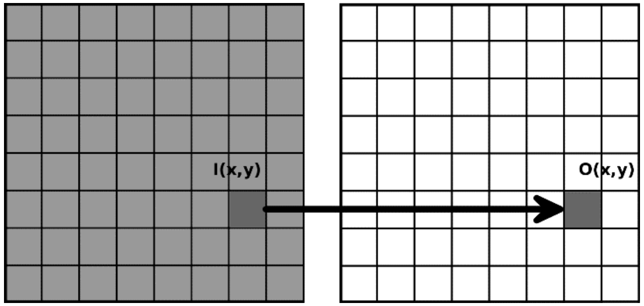
We distinguish 4 categories:



Global operations:

regard whole image to change pixel
E.g.: histogram equalization or Fourier transformation

More in lecture on histograms

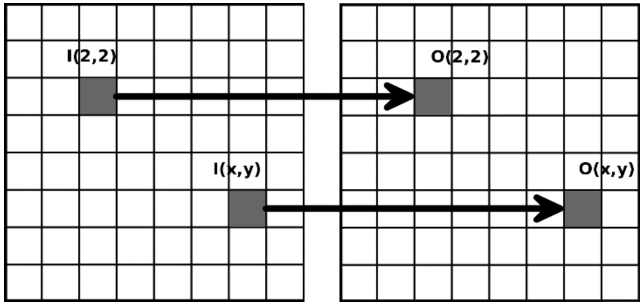


Spatial Operations

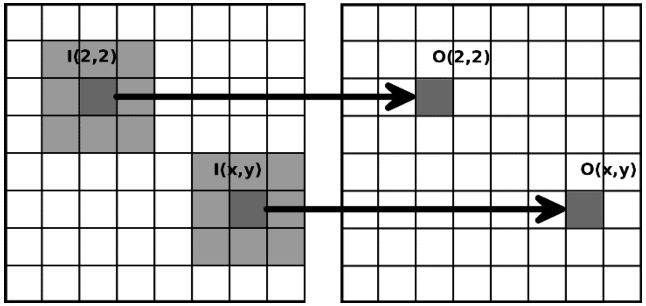
Spatial operations are operations directly applied to the pixels of an image.

We distinguish 4 categories:

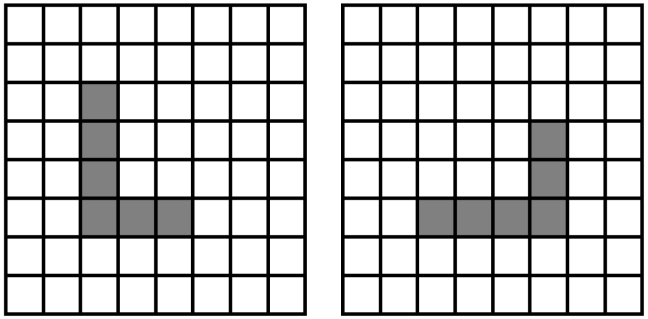
Point operations



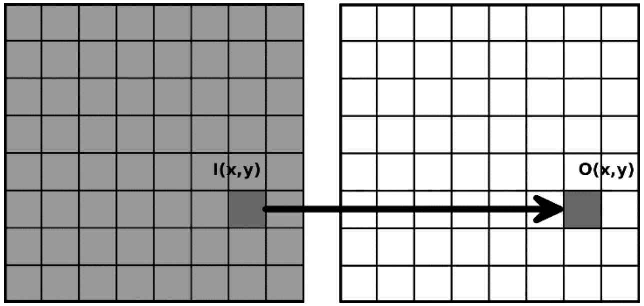
Neighborhood operations:



Geometric transformations



Global operations:



Literature

- Gonzales/Woods: chapter 2, relevant parts

- More on arithmetic and logical operations also here: (older examples, but quite illustrative)
  Image processing learning resources,
  https://homepages.inf.ed.ac.uk/rbf/HIPR2/arthops.htm

Outline

- Part 1: Element-wise vs Matrix Operations
- Part 2: Arithmetic Operations: Addition
- Part 3: Arithmetic Operations: Subtraction
- Part 4: Arithmetic Operations: Multiplication & Division
- Part 5: Set Operations
- Part 6: Logical Operations
- Part 7: Spatial Operations