Images Intensity Transformations

Image Processing

Outline

- Color depth
- Color spaces
- Arithmetical operations on images
- Physically correct filtering

Color depth

Mainly two types for component storage are used

- Integer values
  - [0, 255] range for BYTE
  - [0, 65535] range for WORD (2 bytes)
- Floating point values
  - in RGB it have visible range [0, 1]
  - all values are allowed

Color depth

Problems of integer values

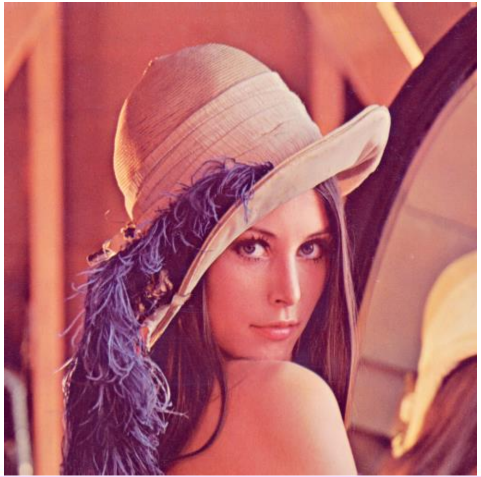
- All operations are rounded up
  - 3 / 2 = 1
- Not always possible to invert
  - 3 / 2 = 1
  - 1 * 2 ≠ 3
- No negative numbers (in general)
  - 0 - 1 = 255 (for BYTE)
- Overflow is possible
  - 255 + 1 = 0 (for BYTE)

Color depth

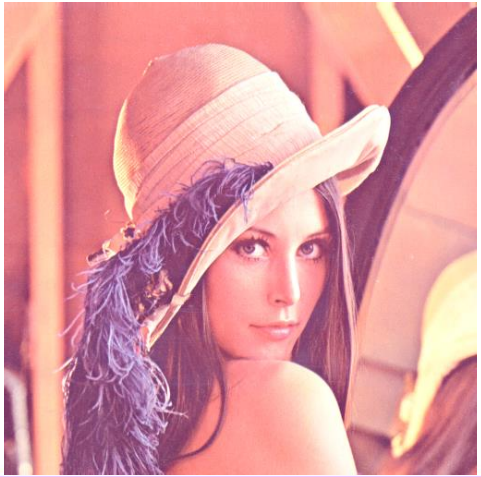
Problems of integer values

- Overflow example
  - I = I + 50

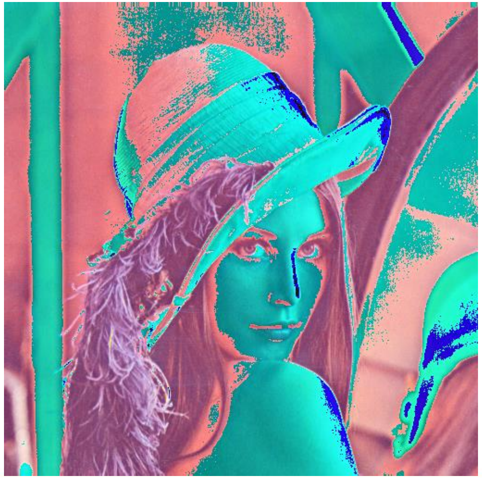
Initial



Correct



Overflow



Color depth

Why do we need floating point values?

- Wider range
- Allows using negative colors
  - for example, for gradients calculation
- Inversibility
  - most of intensity transformations can be losslessly inverted

- Can’t use look up tables (LUT)
  - in general

Color spaces

Is RGB color space suitable for intensity transformations?

- We have to transform all layers
  - different components have different luminance perception weights

- So, we should better use some color space that allows transforming only luminance component while keeping chromaticity components intact (or vice versa)
  - CIE Lab
  - HSV, HSL
  - CIE xyY

Color spaces

Which color space fits most?

- CIE Lab
  - have separate component for luminance
  - uses floating point values
  - complex transformation formula
- HSV, HSL
  - have separate component for luminance
  - uses integer values
  - complex transformation formula
- CIE xyY
  - have separate component for luminance
  - uses floating point values
  - simple matrix transformation to XYZ -> xyY

Color spaces

Which color space fits most?

- CIE xyY
  - have separate component for luminance
  - uses floating point values
  - simple matrix transformation to XYZ and then to xyY
  - is a CIE reference color space
  - Y is luminance of the color

$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$

$$Y = Y$$

$$X_i = \frac{x_i Y_i}{y_i}$$

$$Y_i = Y_i$$

$$Z_i = \frac{(1 - x_i - y_i) Y_i}{y_i}$$

Color spaces

Which color space fits most?

- HSV
- Pros
  - is based on a human perception model
  - can use floating point values
  - V is luminance of the color
  - simple recoloring by modifying H component
- Cons
  - complex transformation formula
  - is not a CIE reference color space

Simple intensity transformations

Histogram transformations

- Range stretching
- Equalization
- Histogram matching
- etc.

- Can be applied to luminance layer only
  - would keep chromaticity intact
- Can be performed using lookup tables (LUT)
  - only in case of integer values

Arithmetical operations on images

Image can be considered as a matrix

- Matrix operations are possible
- Each image layer is a separate matrix
  - can transform only luminance in xyY color space
  - can recolor by transforming hue in HSV

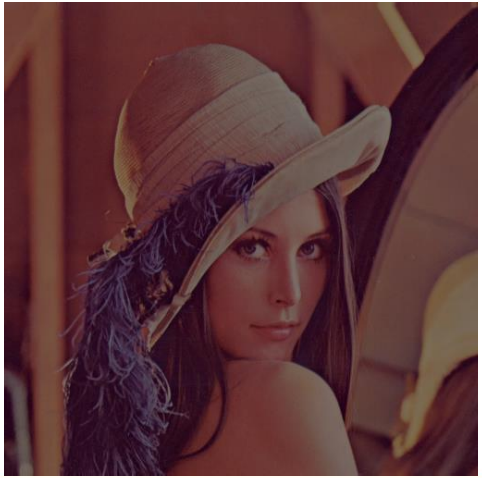
Arithmetical operations on images

Image can be considered as a matrix

- Multiplication by a number
  - $I_{new} = I * mul$
  - either for all layers (RGB, XYZ) of for a single layer (xyY, HSV)

* 0.5

=



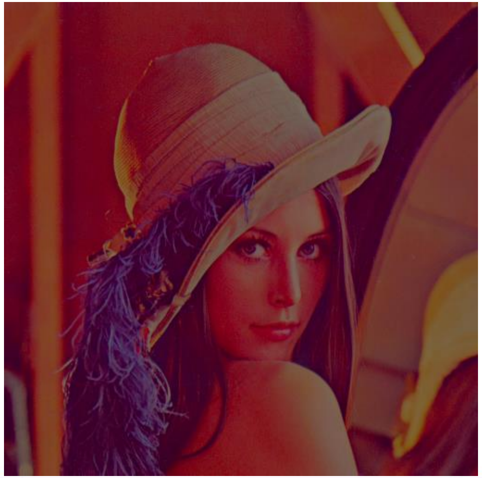
Arithmetical operations on images

Source

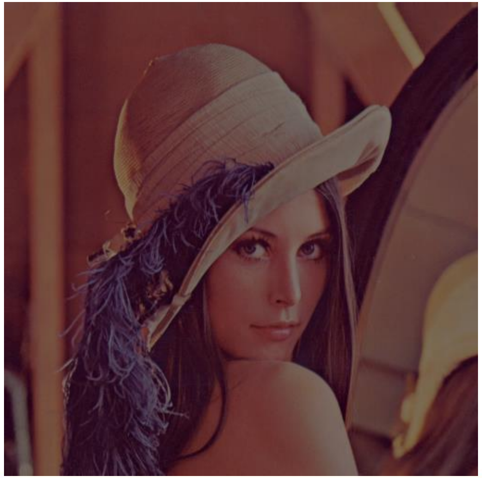
Image can be considered as a matrix

- Multiplication by a number
  - results may depend on a color model

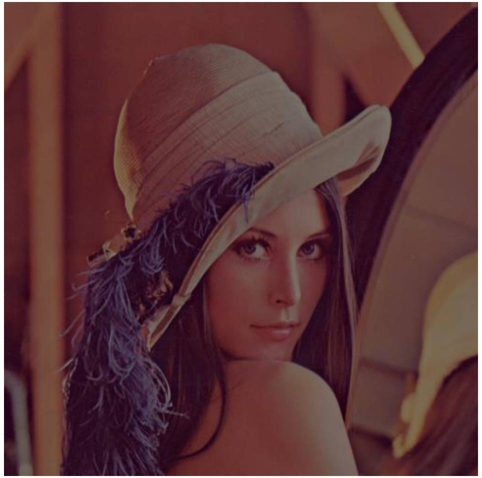
CIE Lab



HSV



CIE xyY

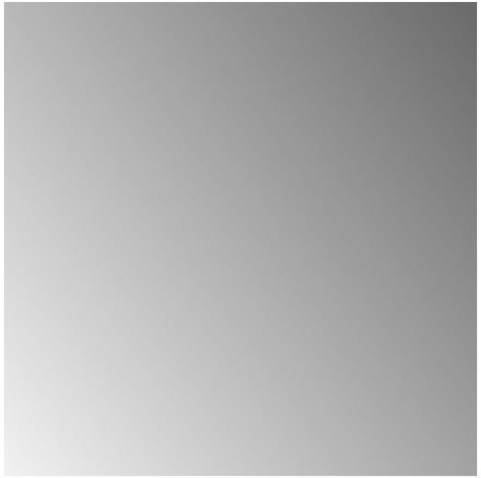


Arithmetical operations on images

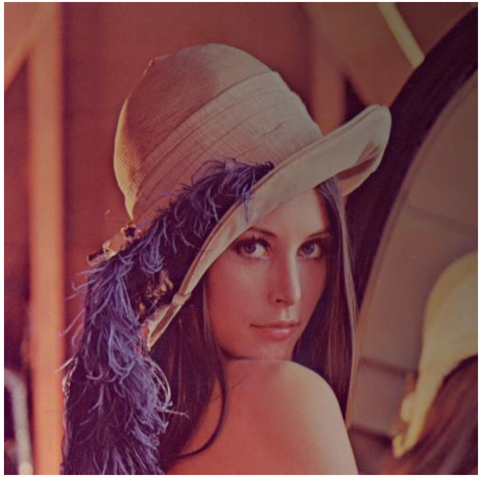
Image can be considered as a matrix

- Multiplication by a matrix
  - $I_{new} = I_1 * I_2$
  - either for all layers (RGB, XYZ) of for a single layer (xyY, HSV)

*



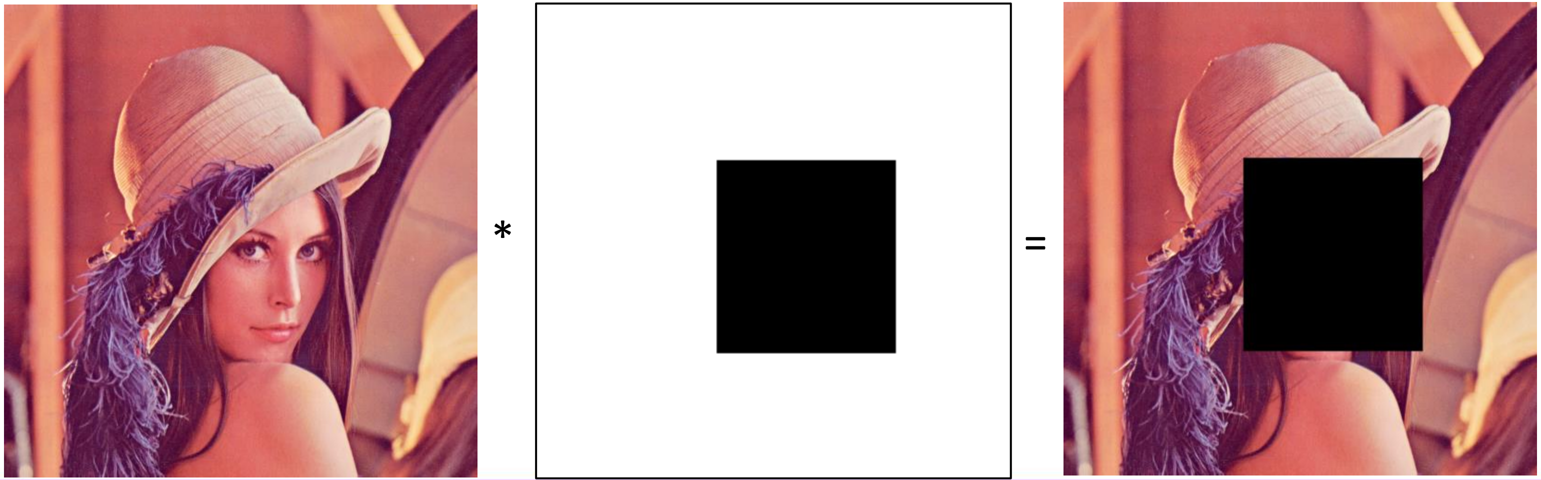
=



Arithmetical operations on images

Image can be considered as a matrix

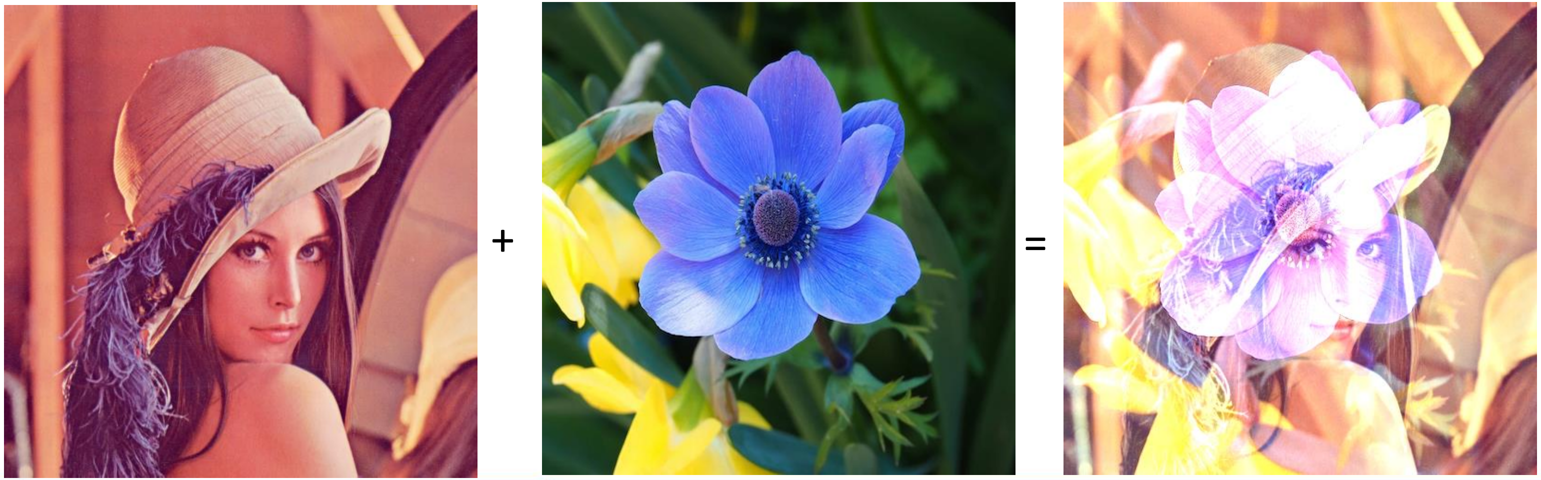
- Multiplication by a matrix
  - masking
  - mask is a binary image



Arithmetical operations on images

Image can be considered as a matrix

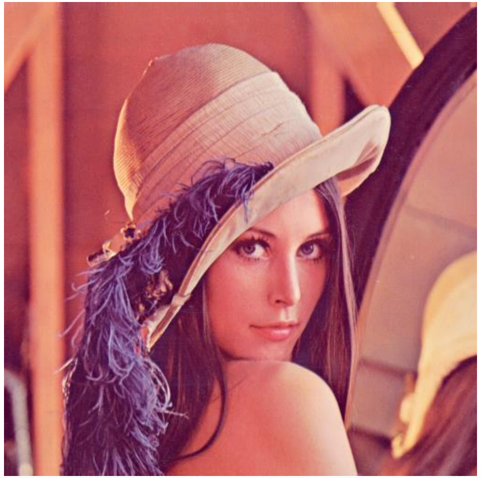
- Summing up images
  - mixing
  - $I_{new} = I_1 + I_2$



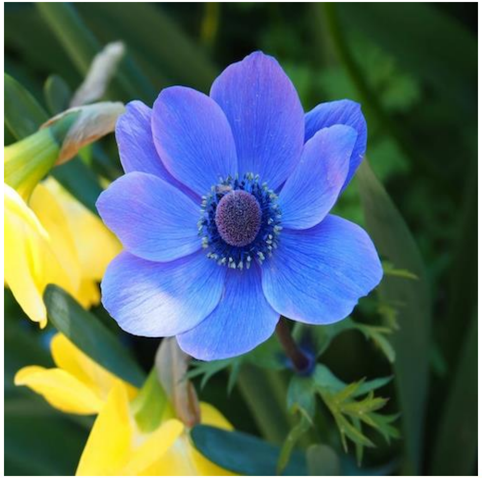
Arithmetical operations on images

Image can be considered as a matrix

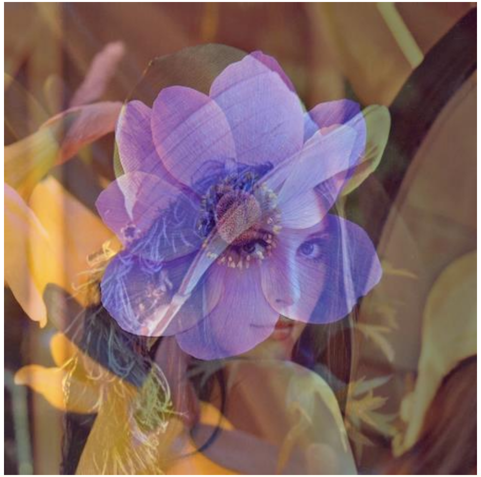
- Summing up images with weights
  - $I_{new} = I_1 * weight_1 + I_2 * weight_2 = I_1 * weight_1 + I_2 * (1 - weight_1)$
  - $I_{new} = I_1 * 0.5 + I_2 * 0.5$



+



=



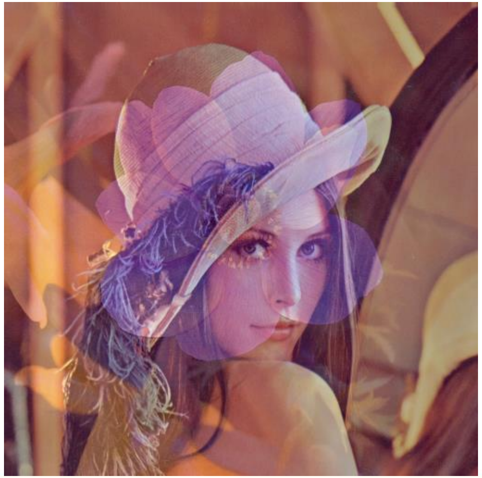
Arithmetical operations on images

Image can be considered as a matrix

- Summing up images with weights
  - $I_{new} = I_1 * weight_1 + I_2 * weight_2 = I_1 * weight_1 + I_2 * (1 - weight_1)$
  - $I_{new} = I_1 * 0.7 + I_2 * 0.3$

+

=



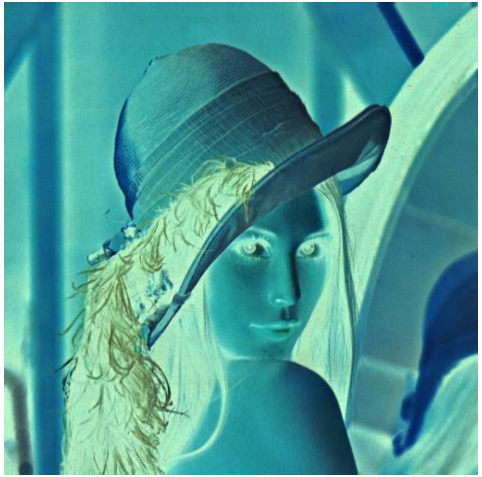
Arithmetical operations on images

Image can be considered as a matrix

- Inversing an image
  - combination of matrix operations
  - $I_{new} = I * (-1) + 1$

* (-1) + 1

=



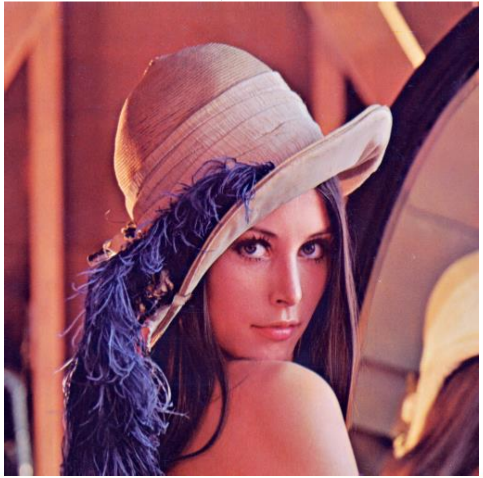
Arithmetical operations on images

Image can be considered as a matrix

- Histogram transformations
  - Dynamic range stretching

$$I_{new} = \left(\frac{I - I_{min}}{I_{max} - I_{min}}\right)^{\alpha}$$

=

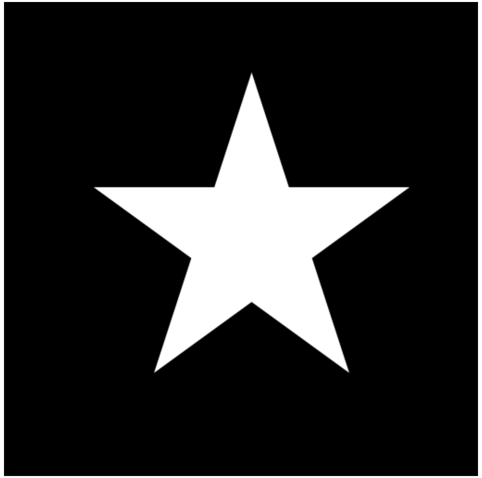


Arithmetical operations on images

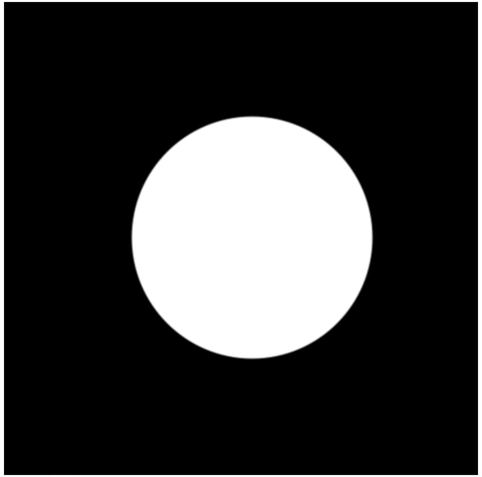
Image can be considered as a matrix

- Logical AND
  - $I_{new} = I_1\ AND\ I_2 = I_1 \wedge I_2 = I_1\ \&\ I_2$

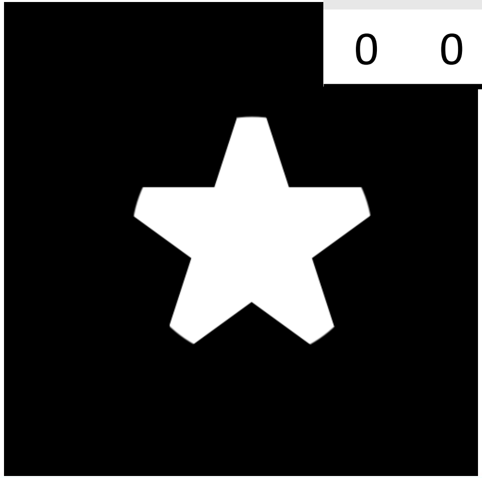
A	B	A∧B
1	1	1
1	0	0
0	1	0
0	0	0



∧



=

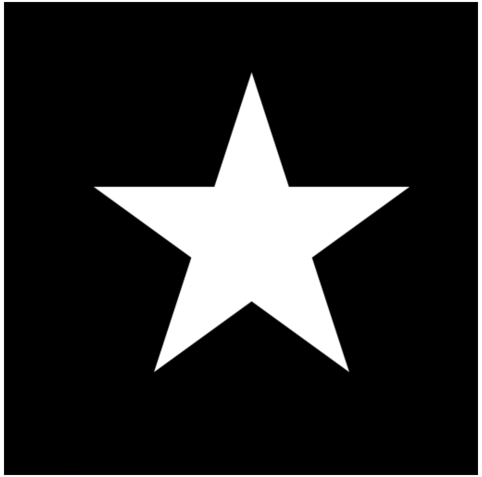


Arithmetical operations on images

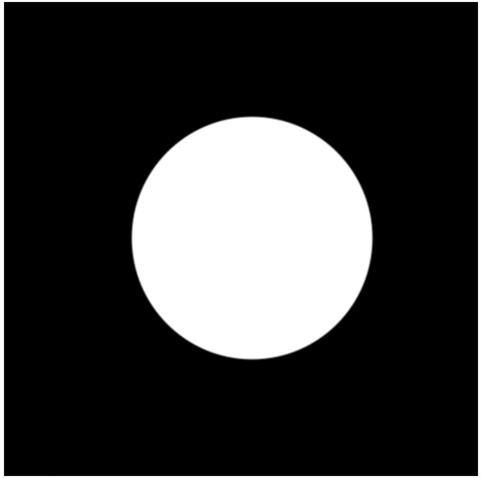
A	B	AVB
1	1	1
1	0	1
0	1	1
0	0	0

Image can be considered as a matrix

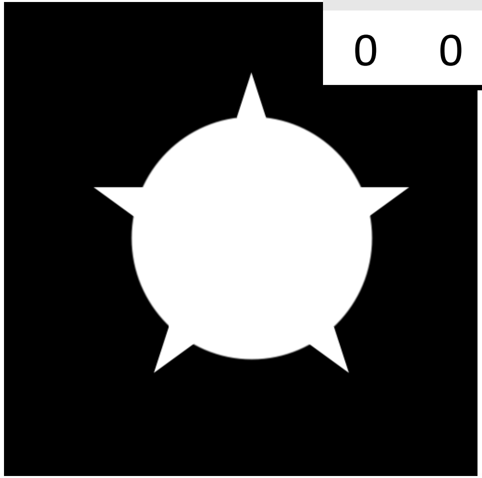
- Logical OR
  - $I_{new} = I_1\ OR\ I_2 = I_1 \vee I_2 = I_1 \mid I_2$



V



=

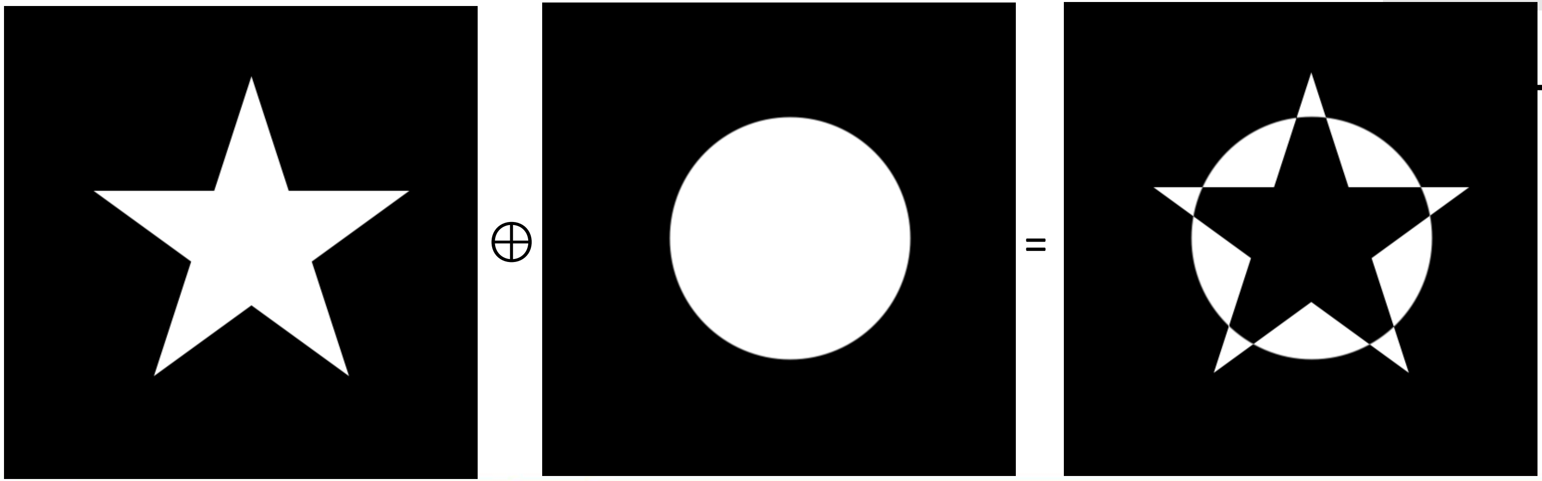


Arithmetical operations on images

Image can be considered as a matrix

- Logical XOR (eXclusive OR)
  - $I_{new} = I_1\ XOR\ I_2 = I_1 \oplus I_2 = I_1$ ^ I_2
  - $I_1\ XOR\ I_2\ XOR\ I_2 = I_1$

A	B	A⊕B
1	1	0
1	0	1
0	1	1
		0

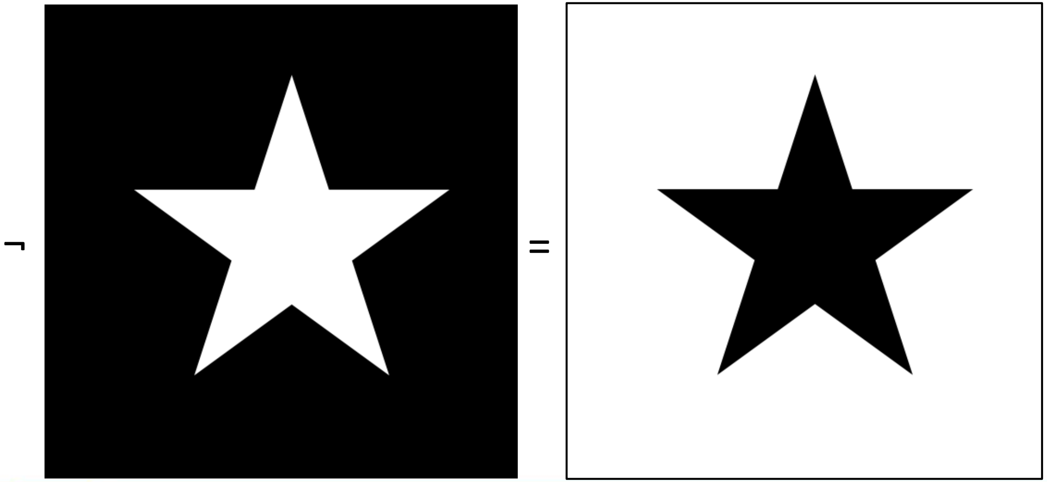


Arithmetical operations on images

Image can be considered as a matrix

- Logical NOT
  - $I_{new} = NOT\ I = \neg I = \bar{I} = !\,I$

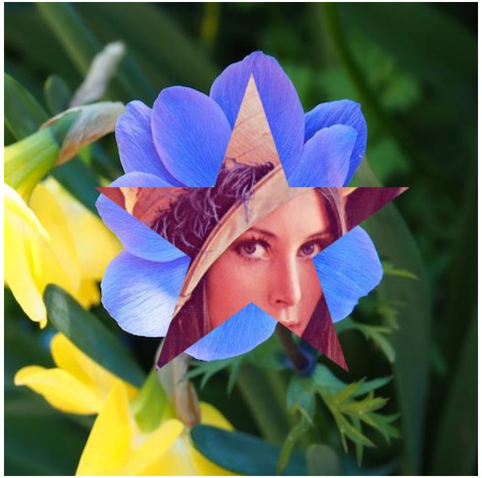
A	¬A
1	0
0	1



Arithmetical operations on images

Image can be considered as a matrix

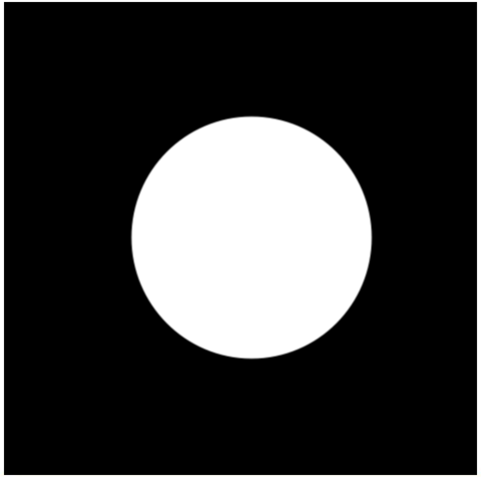
- Image can be processed with mask
  - $I_{new} = I_1 * \text{mask} + I_2 * (\text{NOT mask})$

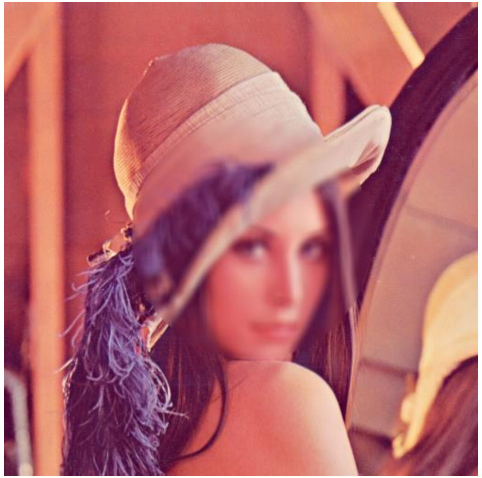


Arithmetical operations on images

Image can be considered as a matrix

- Image can be processed with mask
  - blur with the mask
  - $I_{new} = I_{blur}$ * mask + I * (NOT mask)

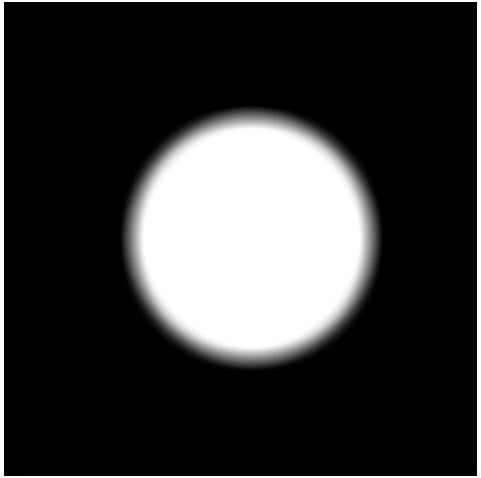


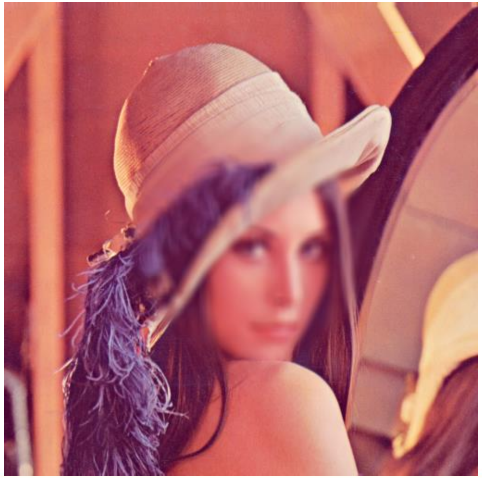


Arithmetical operations on images

Image can be considered as a matrix

- Image can be processed with floating point values mask
  - blur with the mask
  - $I_{new} = I_{blur} * mask + I * (1 - mask)$

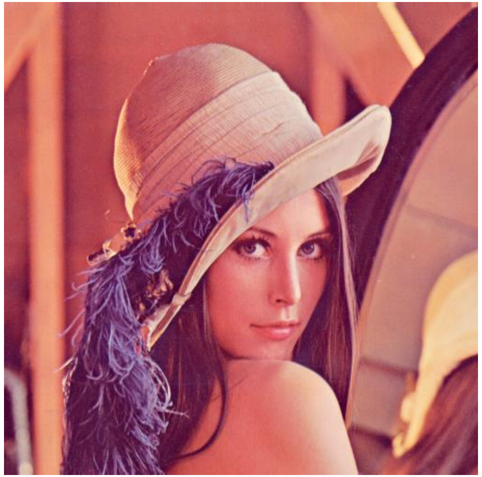


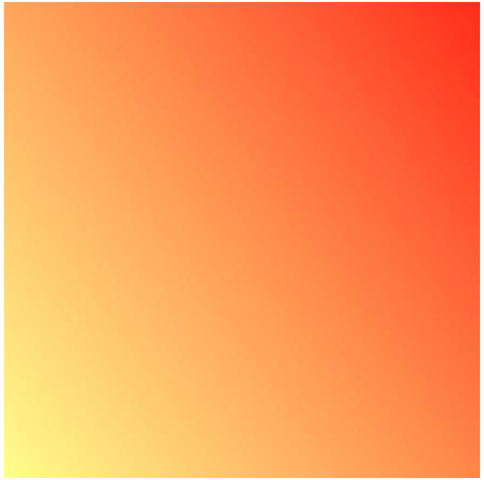


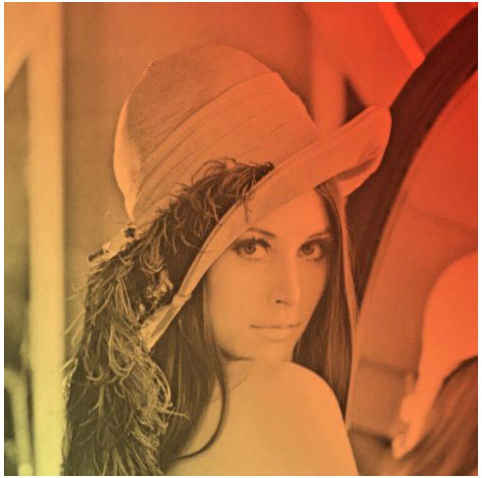
Arithmetical operations on images

Image can be considered as a matrix

- Image can be processed by layers
  - replace H and S layers of image in HSV
  - keep V intact



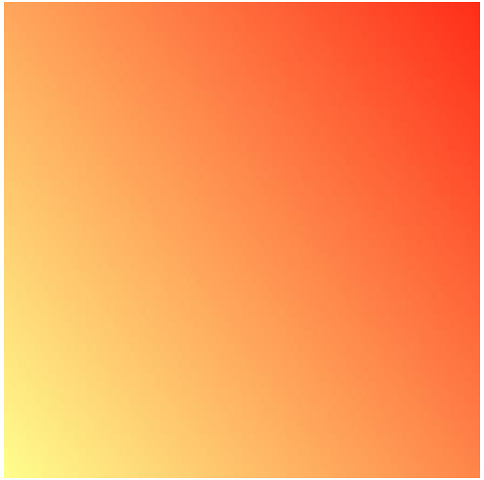


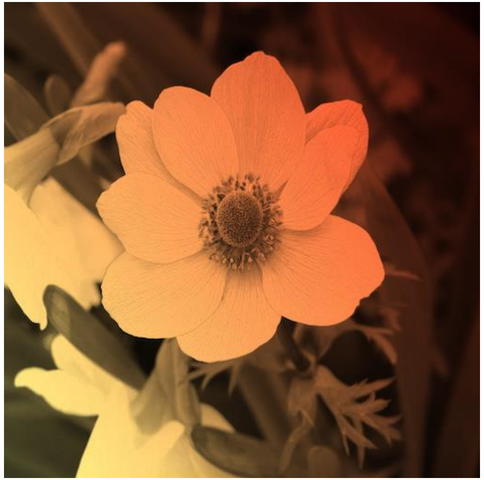


Arithmetical operations on images

Image can be considered as a matrix

- Image can be processed by layers
  - replace H and S layers of image in HSV
  - keep V intact

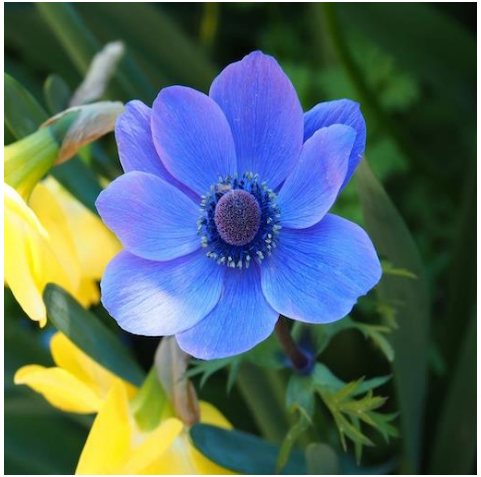


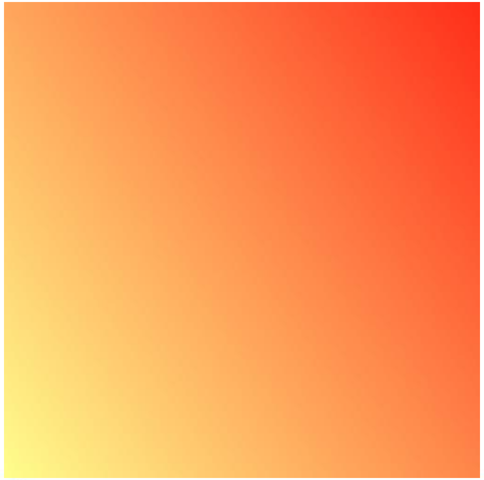


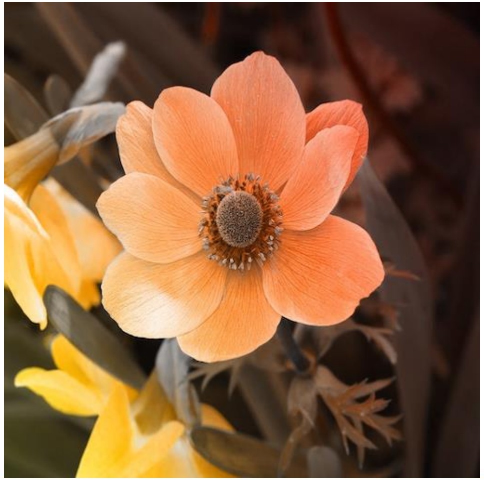
Arithmetical operations on images

Image can be considered as a matrix

- Image can be processed by layers
  - replace H layer of image in HSV
  - keep S and V intact







Physically correct filtering

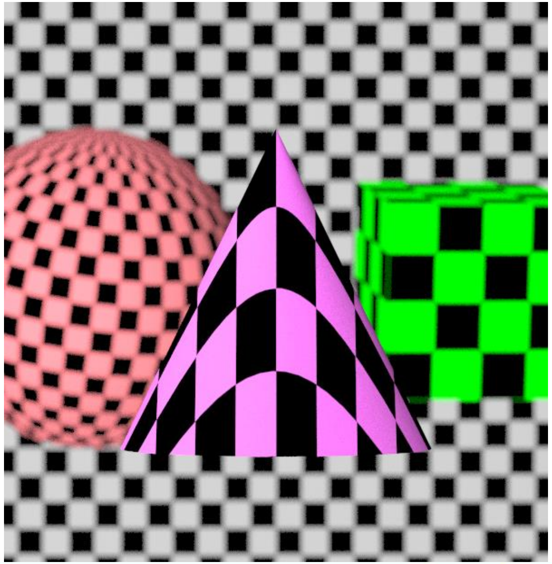
The requirement conditions for physically correct filter

- Should use physical units to store color components
  - radiance / luminance
  - irradiance / illuminance
- Global image luminance should be kept intact
  - luminance can be only transferred
  - it is not allowed to add or remove luminance
    - if it doesn’t have the physical explanation
- Local image luminance should be kept intact
  - luminance can be only transferred within small area

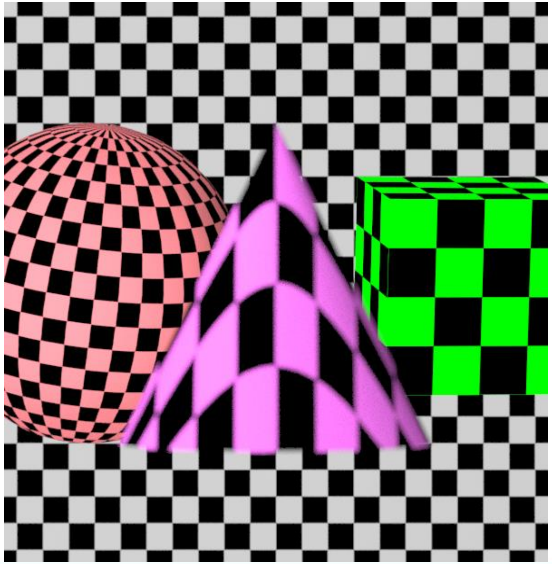
Defocusing

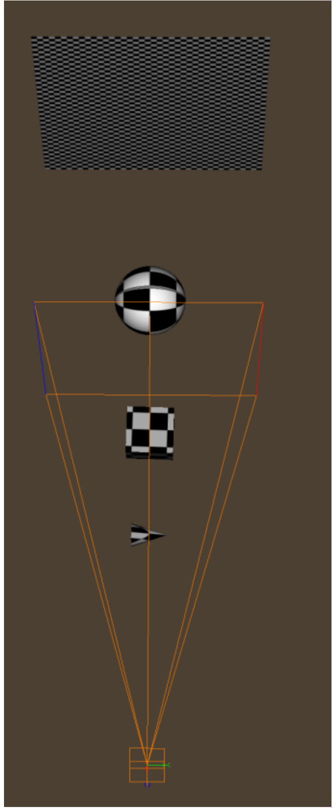
Simulate real lens performance

Focus on near object



Focus on far object





Defocusing

Image forming

Lens

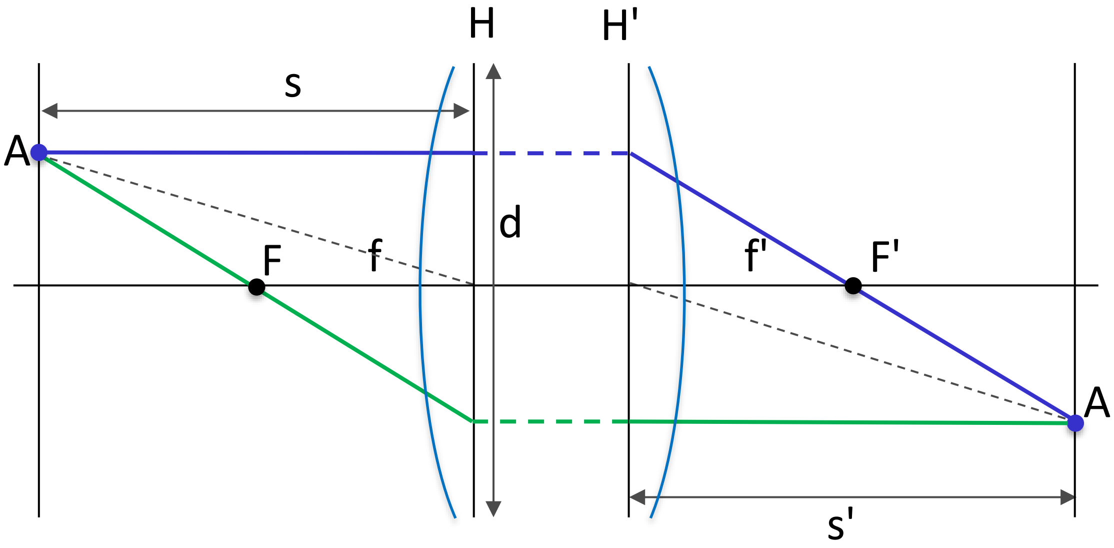
- f' – focal distance
- F – focal points
- H – main planes

A - Object

- s – object distance

A’ – Image

- s' – image distance



$$\frac{1}{s}+\frac{1}{s'}=\frac{1}{f'}$$

Defocusing

Depth of field

Lens

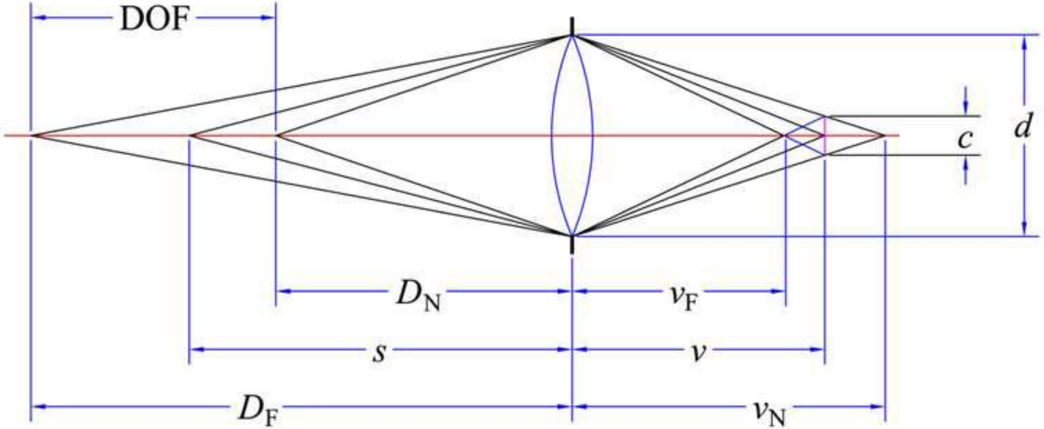
- f' – focal distance
- d – entrance pupil size

Object

- v – image distance
- c – threshold

Image

- s – object distance
- D_n – near DOF
- D_f – far DOF



$$D_n = \frac{sf^2}{f^2 + Nc(s - f')}$$

$$D_f = \frac{sf^2}{f^2 - Nc(s - f')}$$

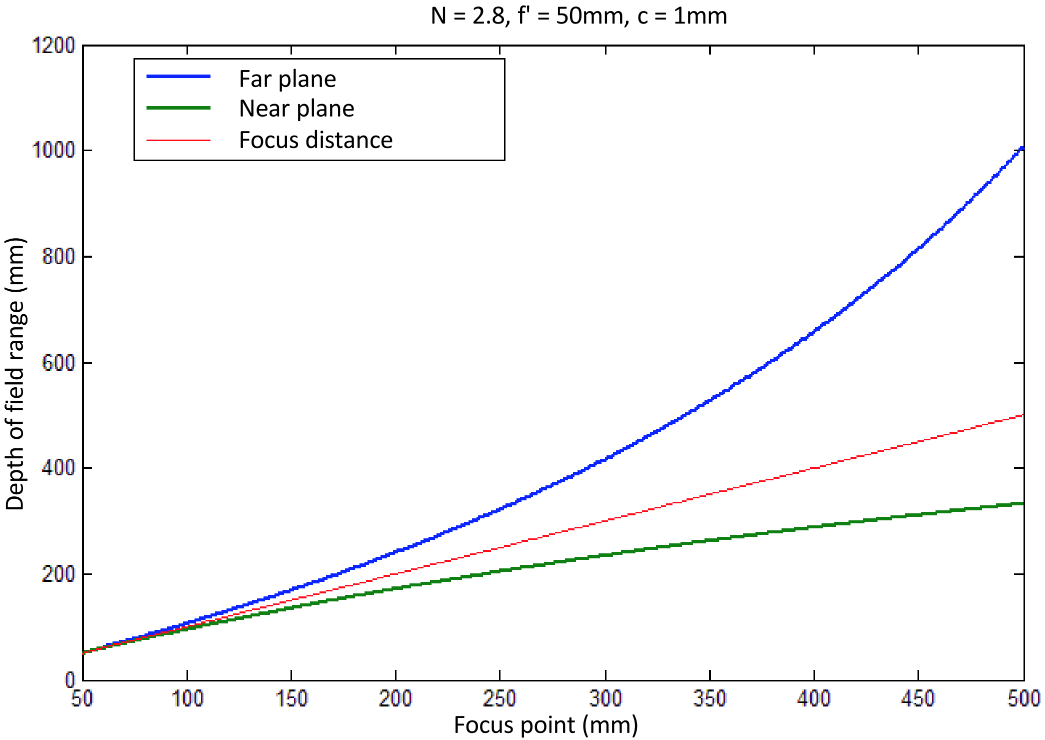
$$N = \frac{f'}{d}$$

$$\frac{1}{s} + \frac{1}{v} = \frac{1}{f'}$$

Defocusing

Depth of field

Range



Defocusing

Modeling

Lens

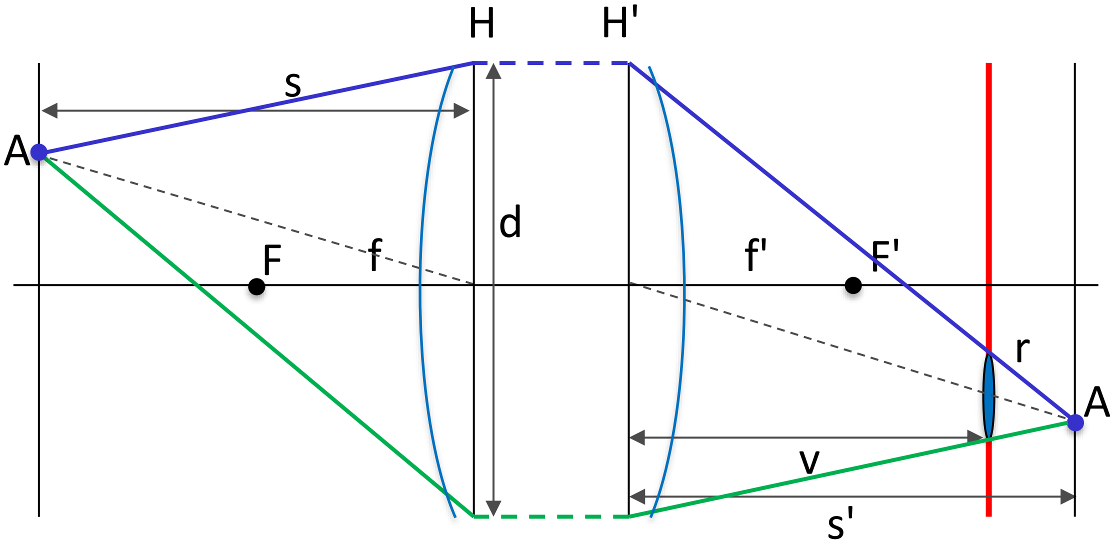
- f' – focal distance
- d – entrance pupil
- H – main planes

A - Object

- s – object distance

A’ – Image

- v – sensor distance
- r – defocused spot radius
- s' – image distance



$$r = \frac{d}{2} \cdot \left| \frac{v(s - f')}{sf'} - 1 \right|$$

$$\frac{1}{s} + \frac{1}{s'} = \frac{1}{f'}$$

Defocusing

Modeling

Lens

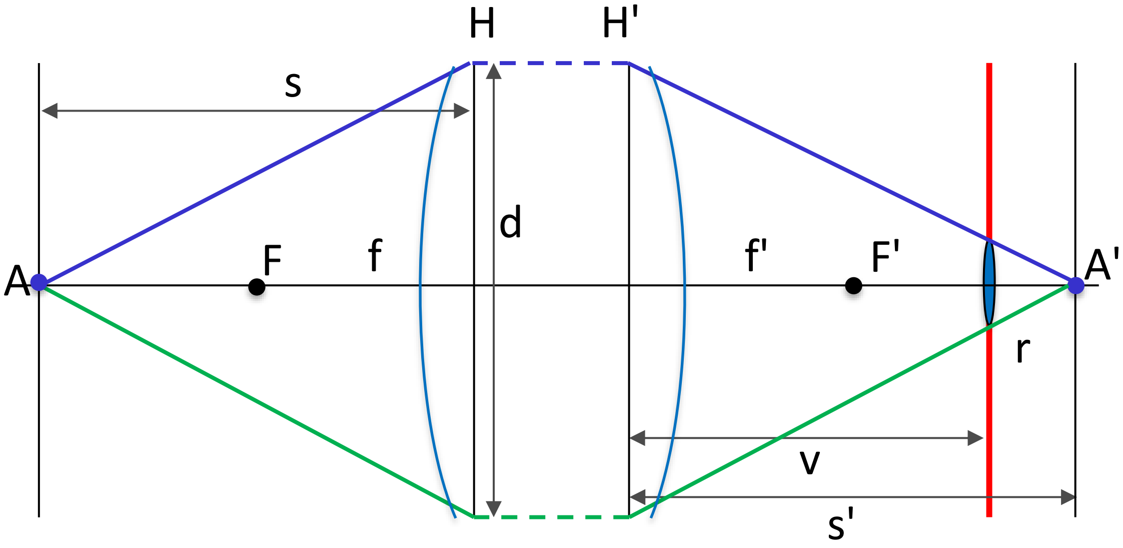
- f' – focal distance
- d – entrance pupil
- H – main planes

A - Object

- s – object distance

A’ – Image

- v – sensor distance
- r – defocused spot radius
- s' – image distance



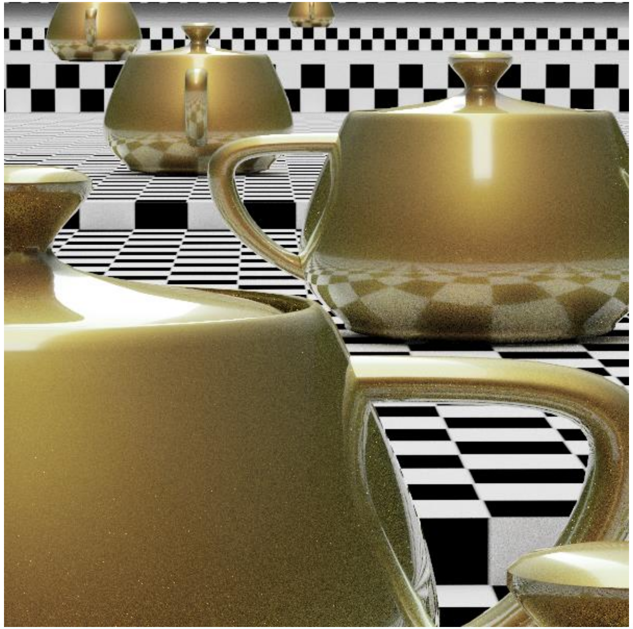
$$r = \frac{d}{2} \cdot \left| \frac{v(s - f')}{sf'} - 1 \right|$$

$$\frac{1}{s} + \frac{1}{s'} = \frac{1}{f'}$$

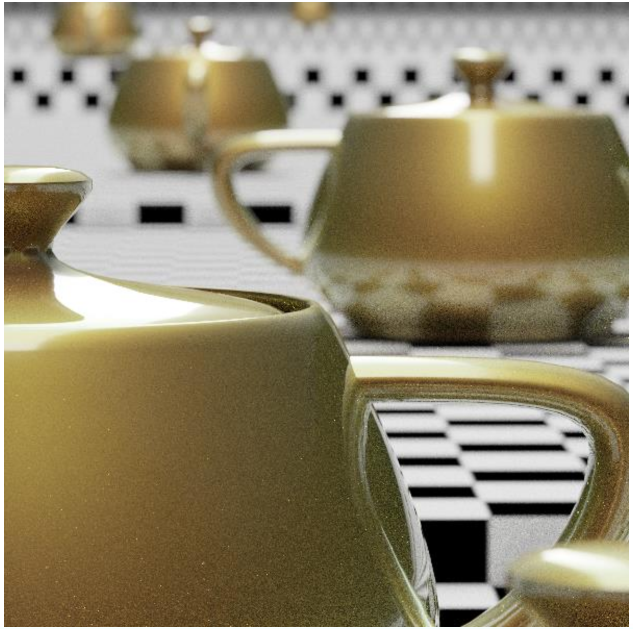
Defocusing

- Focal distance 882mm

Original image



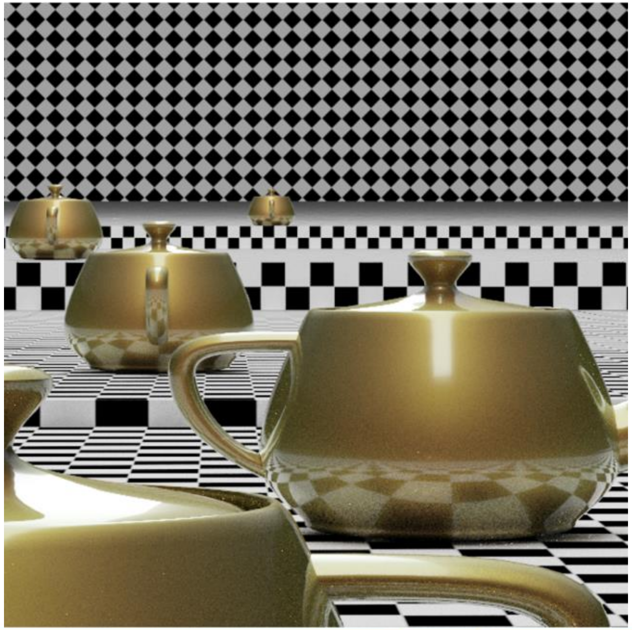
Defocused image



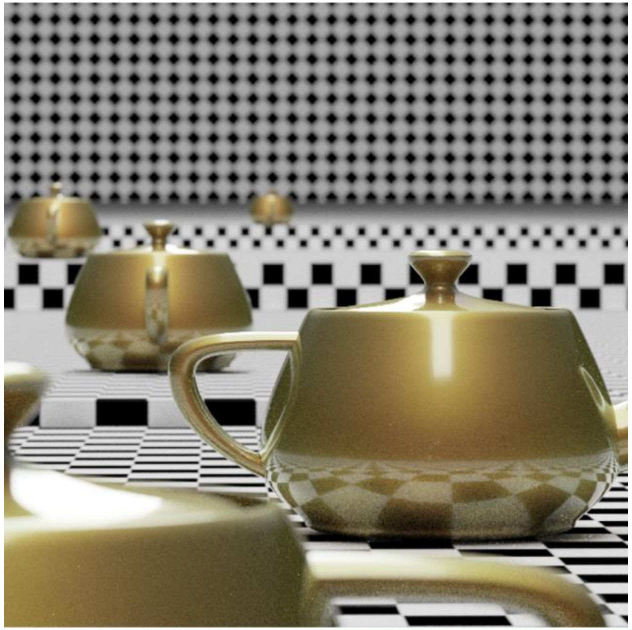
Defocusing

- Focal distance 2076mm

Original image



Defocused image



To conclude

You should consider color depth

- Each one has pros and cons

You should consider color space

- Different transformations require different model

You should consider processed data type

- Some data may have additional correctness criteria

THANK YOU FOR YOUR TIME!

ITsMOre than a UNIVERSITY

Andrei Zhdanov
adzhdanov@itmo.ru