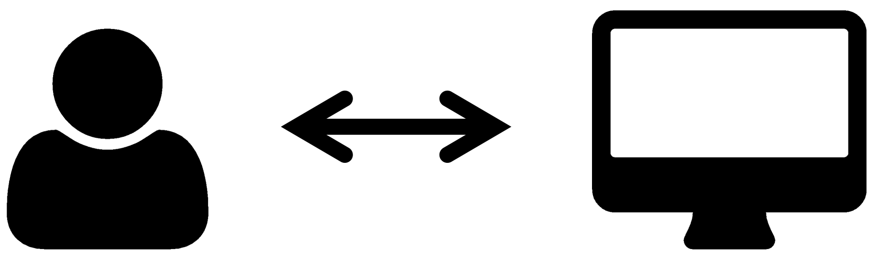
Image correction

02

Image Processing
Dr. Márton Szemenyei
Associate Professor
2024

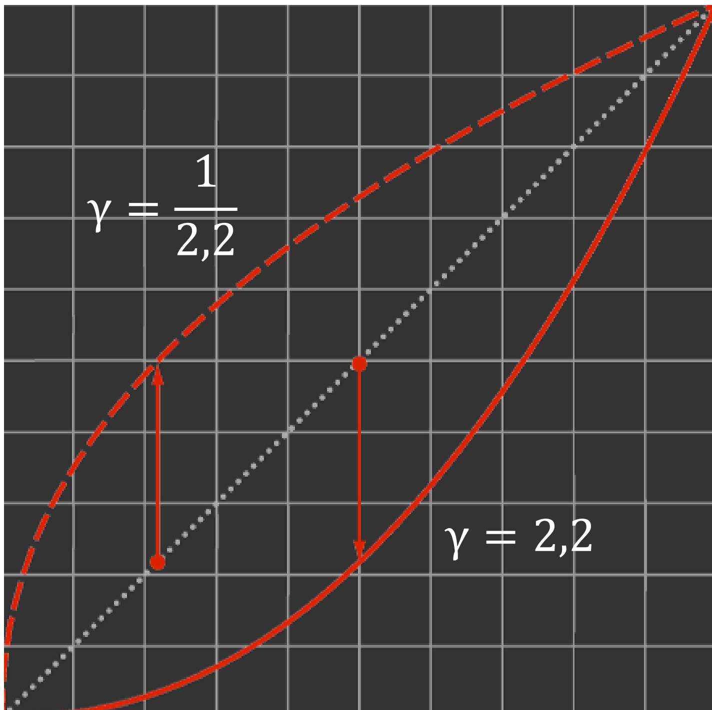
Goals



One good for humans and another for algorithms

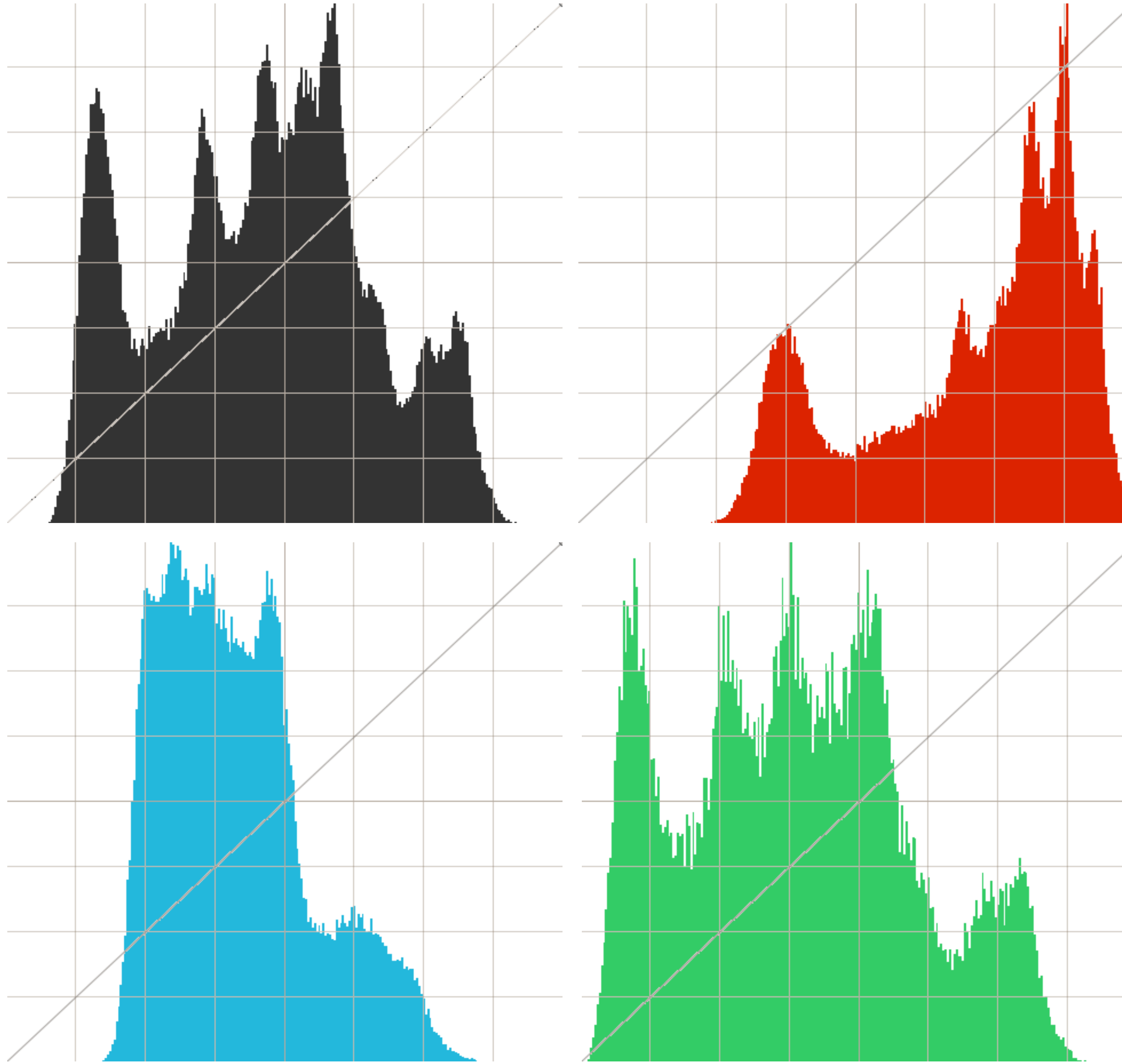
Contrast, brightness

Gamma correction



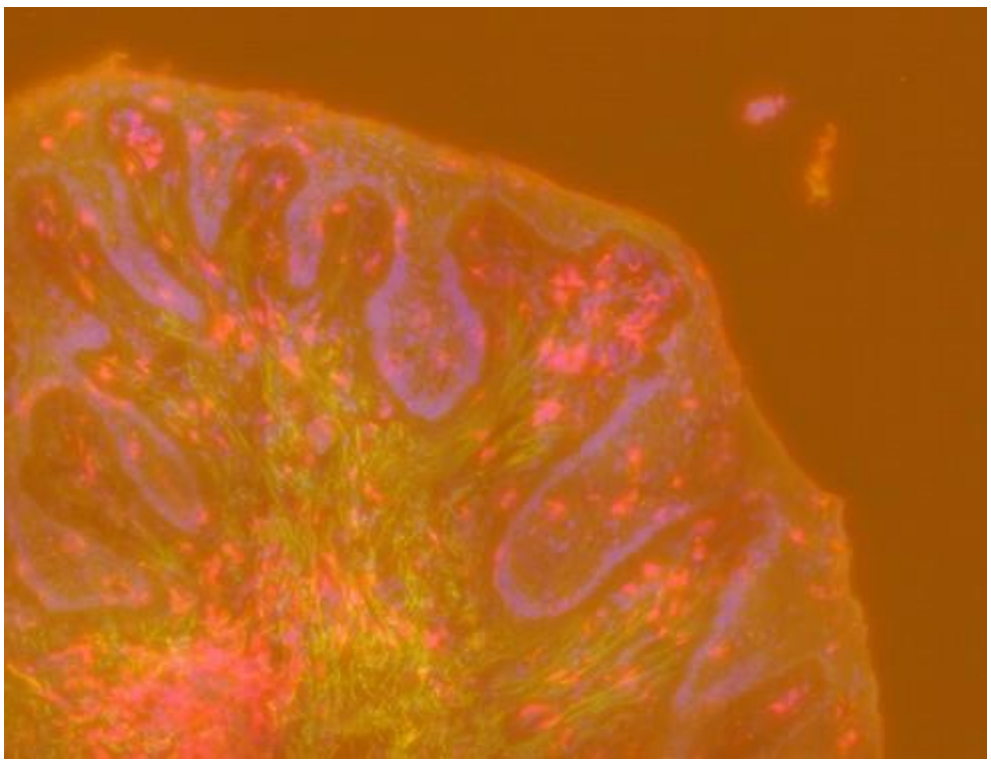
$$I = V_s^{\gamma}$$

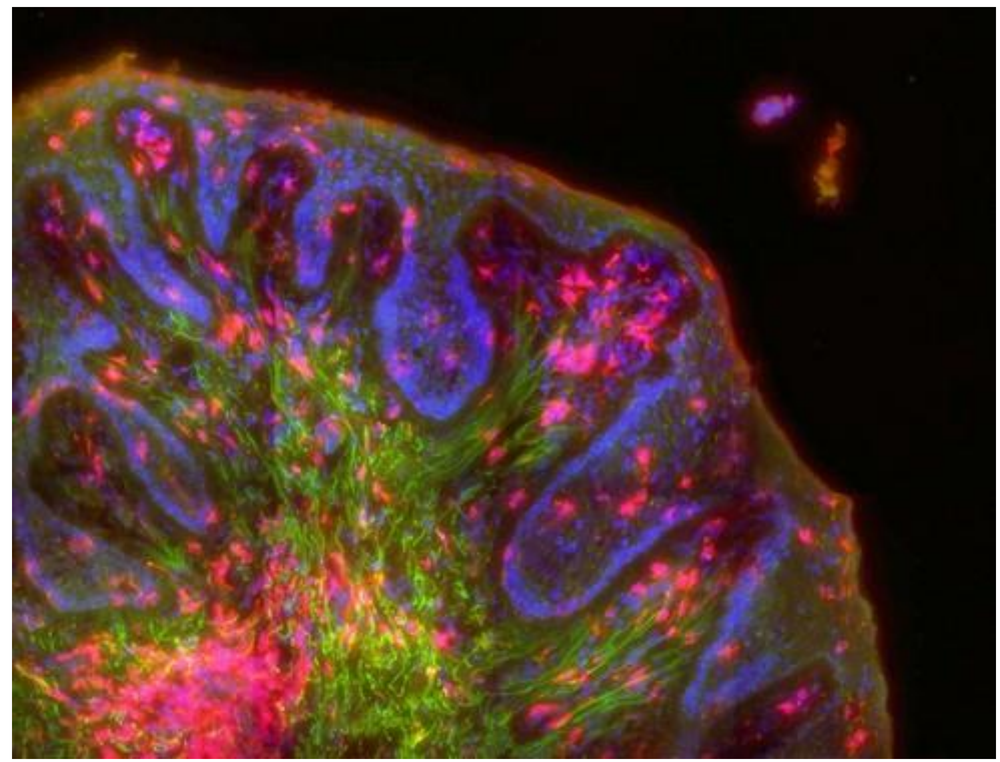
Histogram



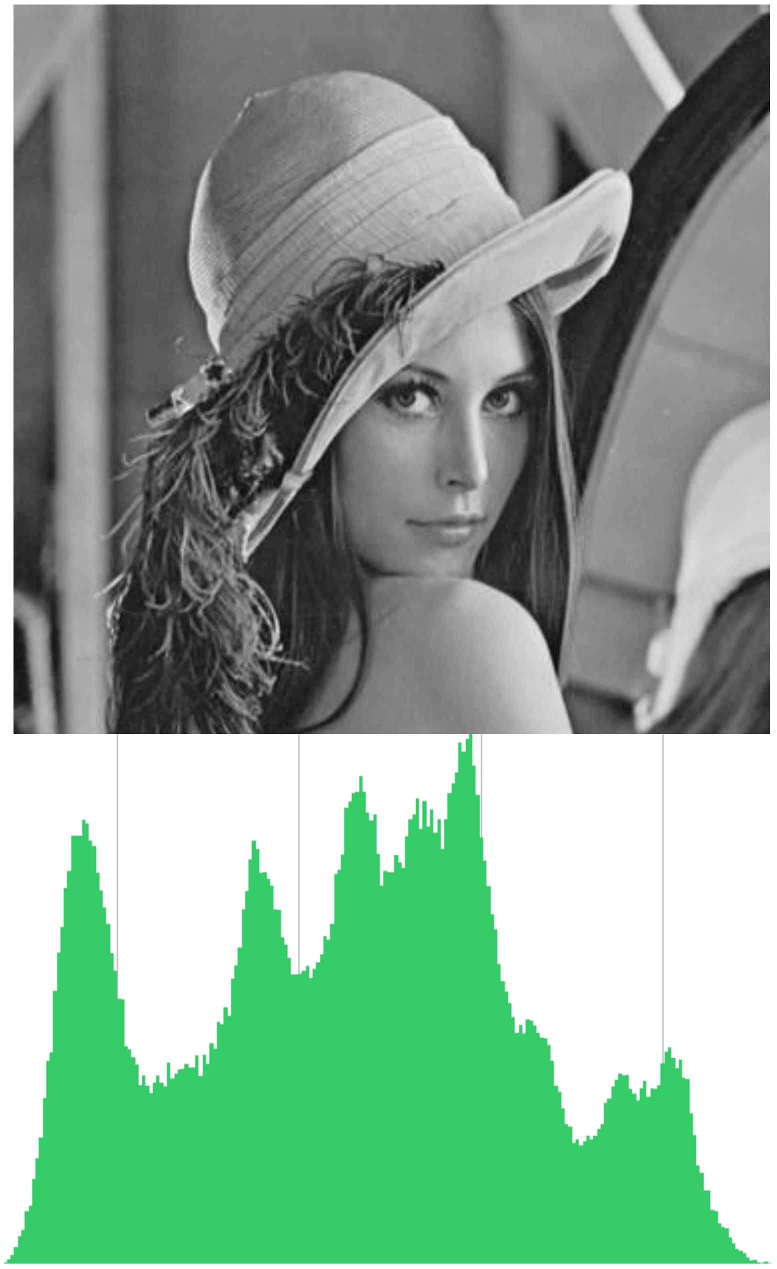
Histogram transformation

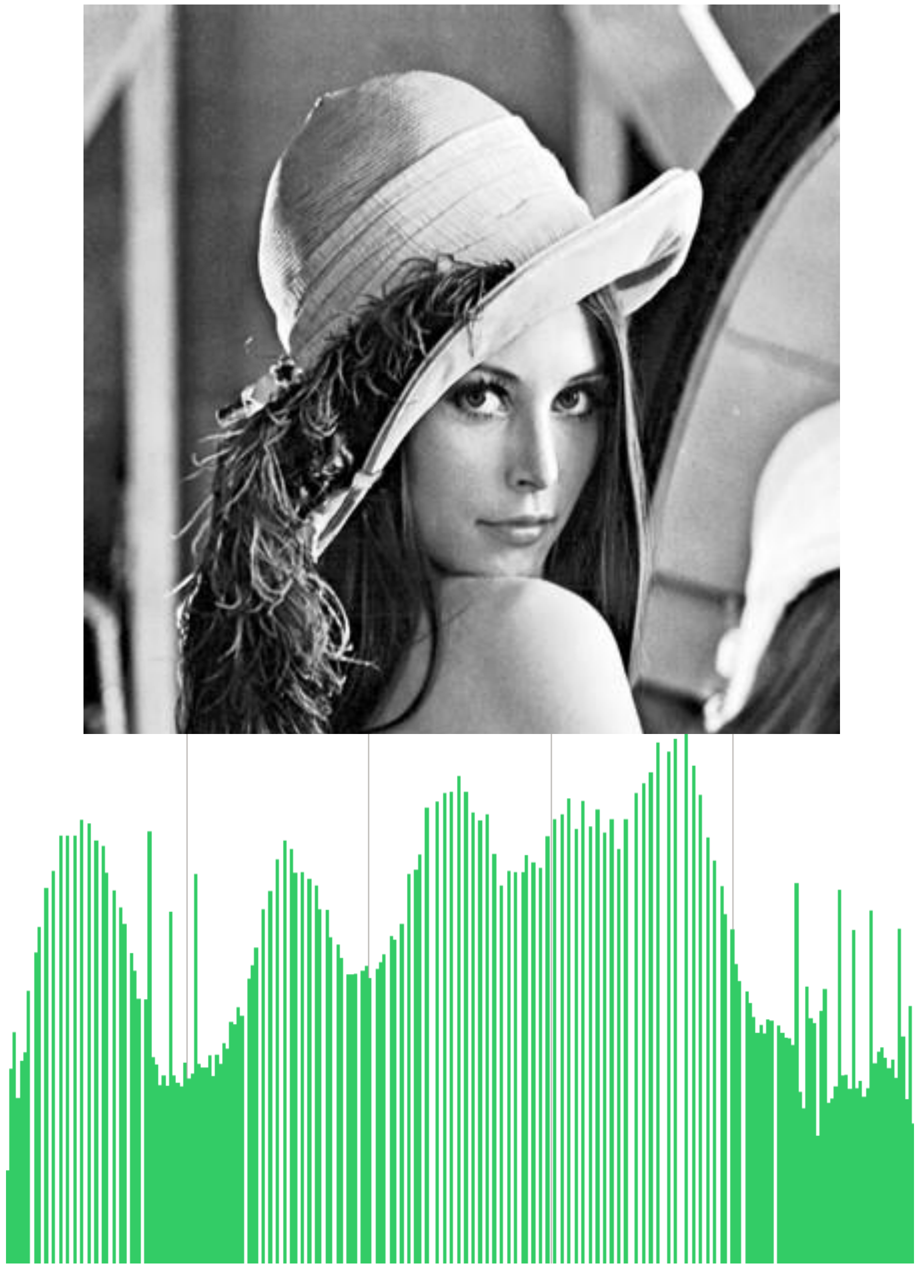
Histogram transformation



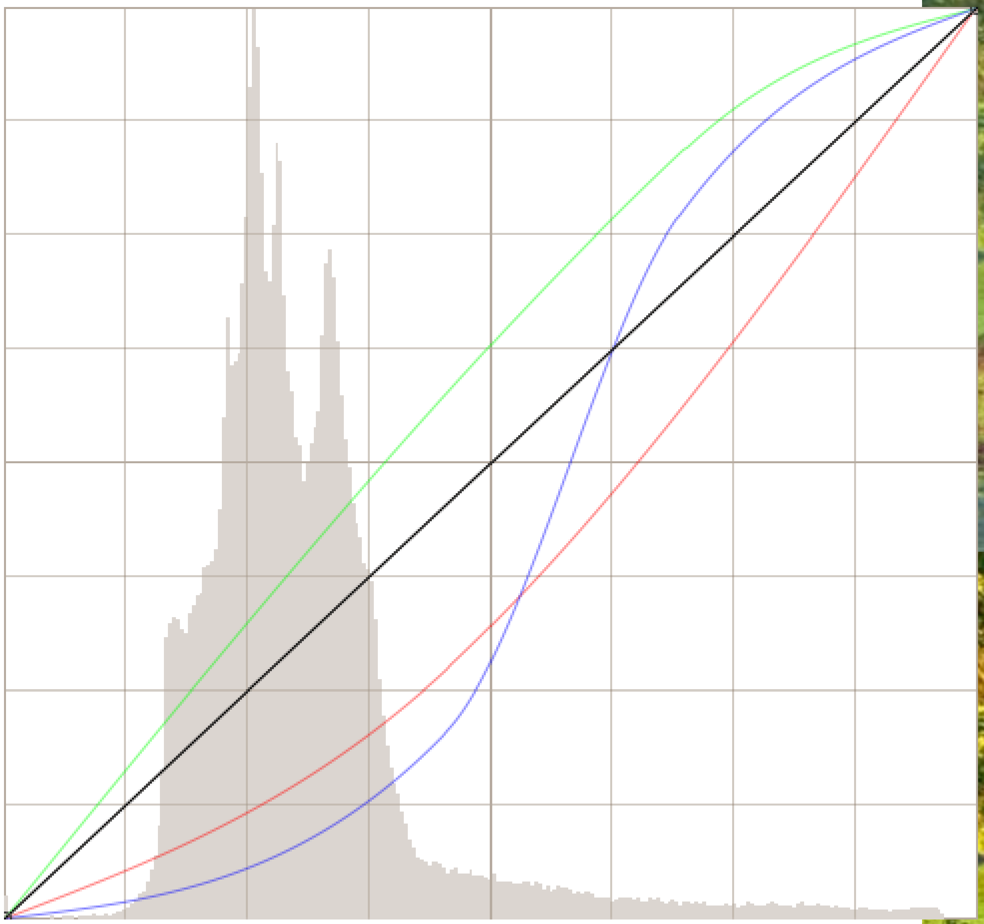


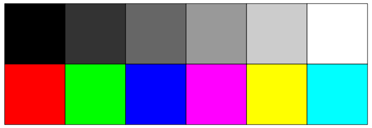
Histogram equalisation





Colour correction

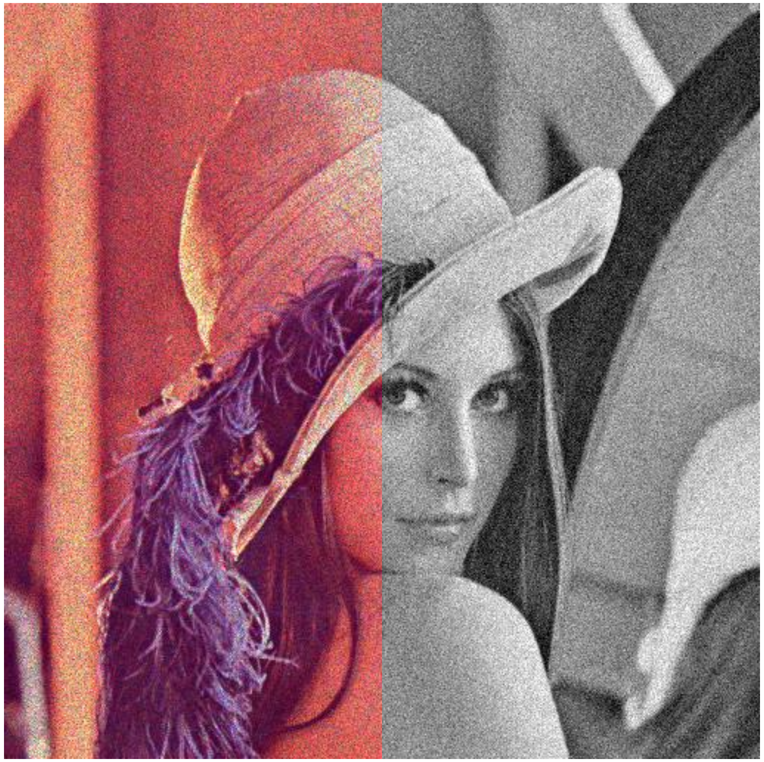




Screenshots

Gauss noise

Salt and pepper noise

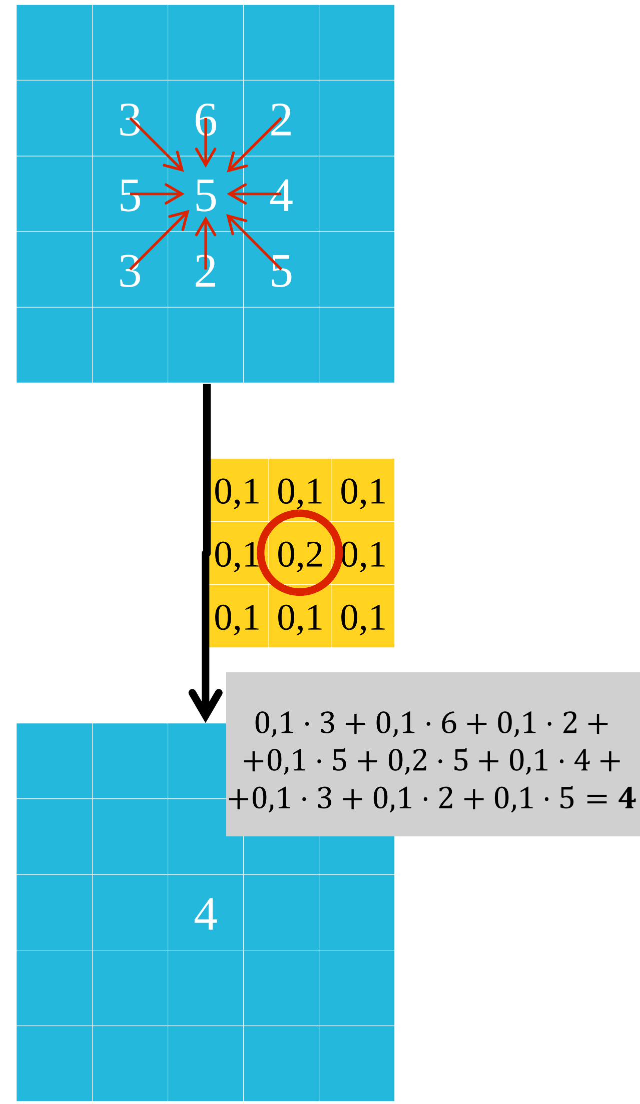


Convolution

$$I_o = k * I_i$$

$$(k * I)(x, y) = \sum_{u=-n}^{n} \sum_{v=-n}^{n} k\,(u, v) \cdot I(x - u, y - v)$$

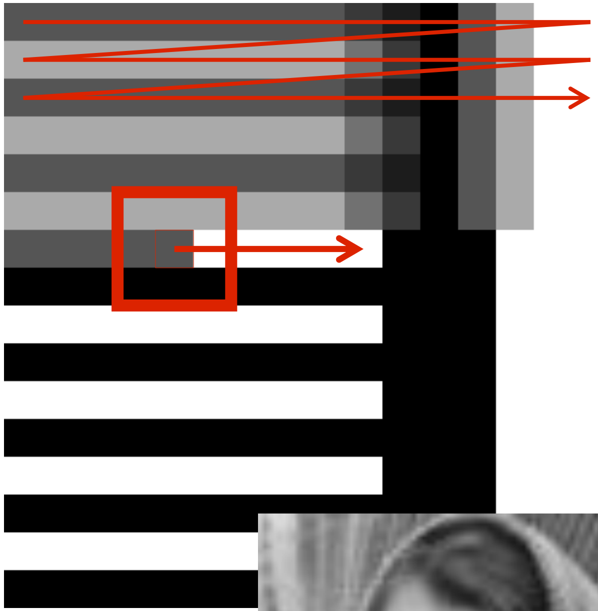
$$(f * g)(x, y) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f\,(u, v) \cdot g(x - u, y - v)$$



Simple averaging

$$p'_{x,y} = \frac{\sum_{u=-1}^{1} \sum_{v=-1}^{1} I\,(x-u, y-v)}{9}$$

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



$$p'_{x,y} = \frac{\sum_{u=-n}^{n} \sum_{v=-n}^{n} I\,(x-u, y-v)}{(2n+1)^2}$$

$$p'_{x,y} = \frac{\sum_{u=-n}^{n} \sum_{v=-n}^{n} k_{u,v} \cdot I(x-u, y-v)}{\sum_{u=-n}^{n} \sum_{v=-n}^{n} k_{u,v}}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

FPGA!

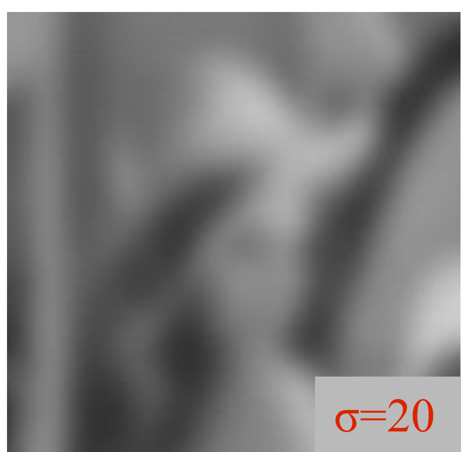
Gaussian filtering

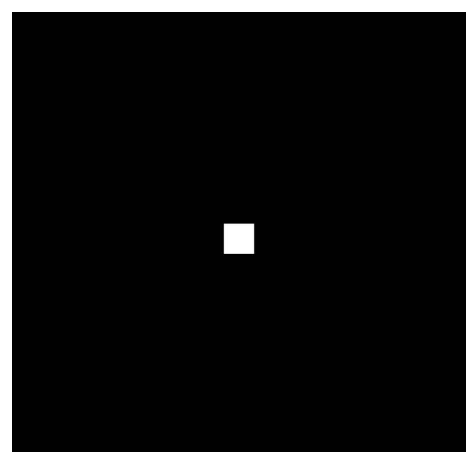
$$k(x, y, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

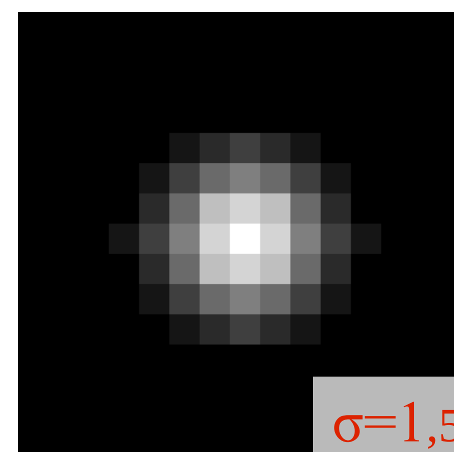
$$\begin{bmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

$\sigma = 0{,}6$

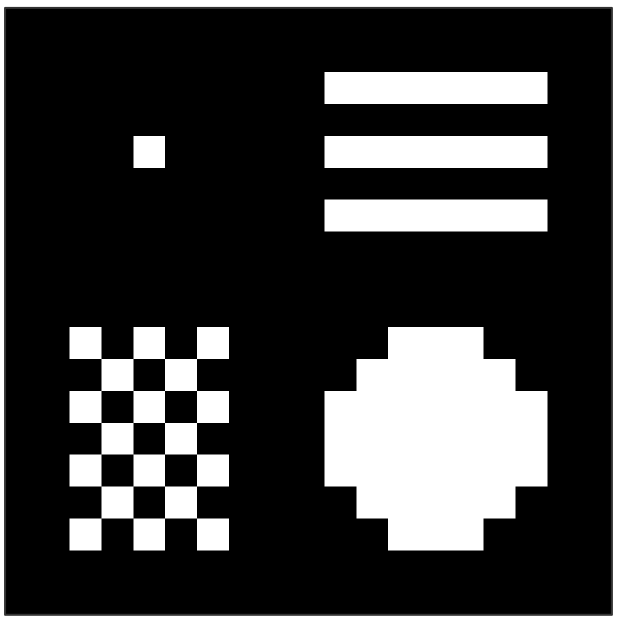
Kernel size: 3σ

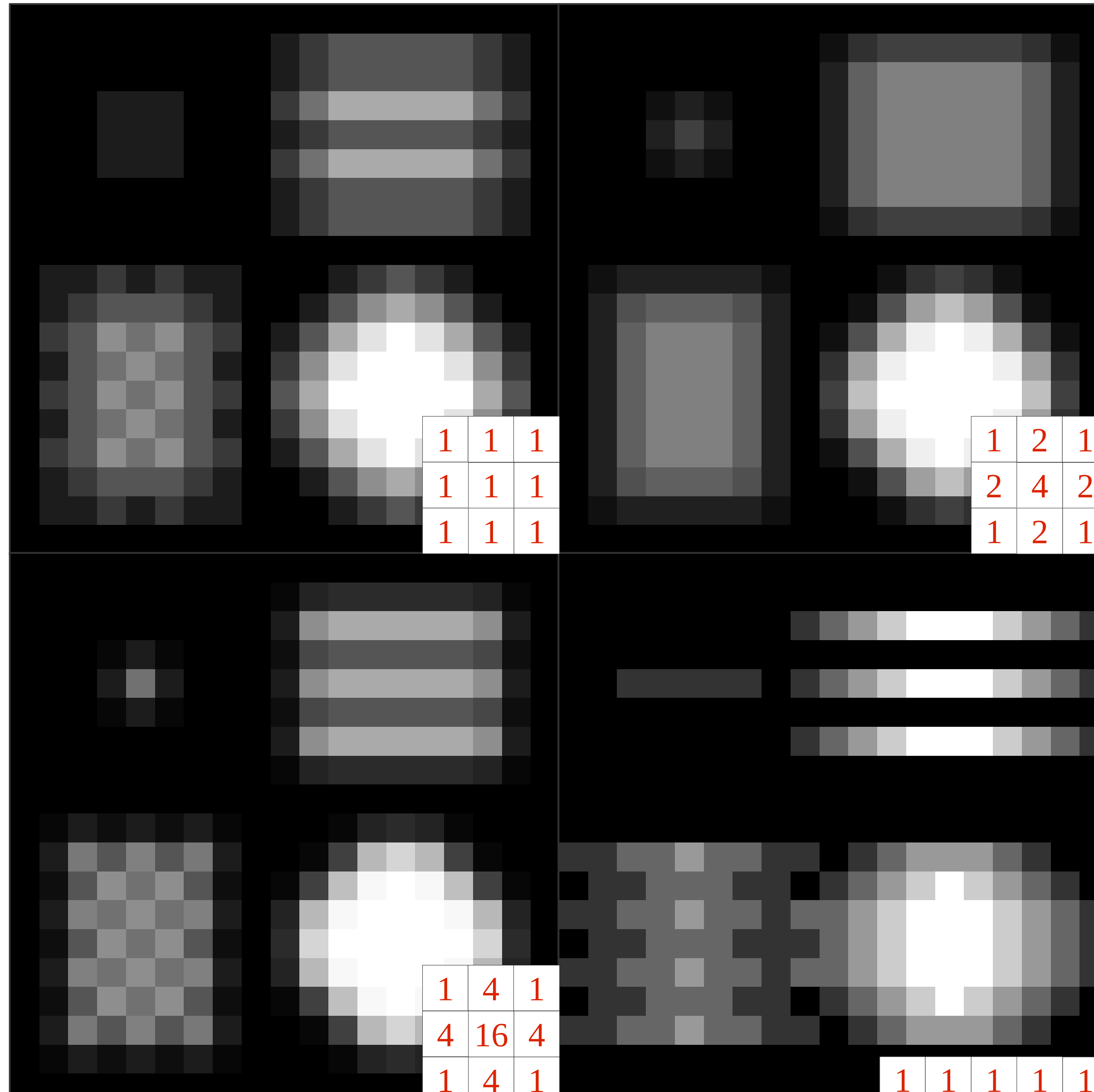






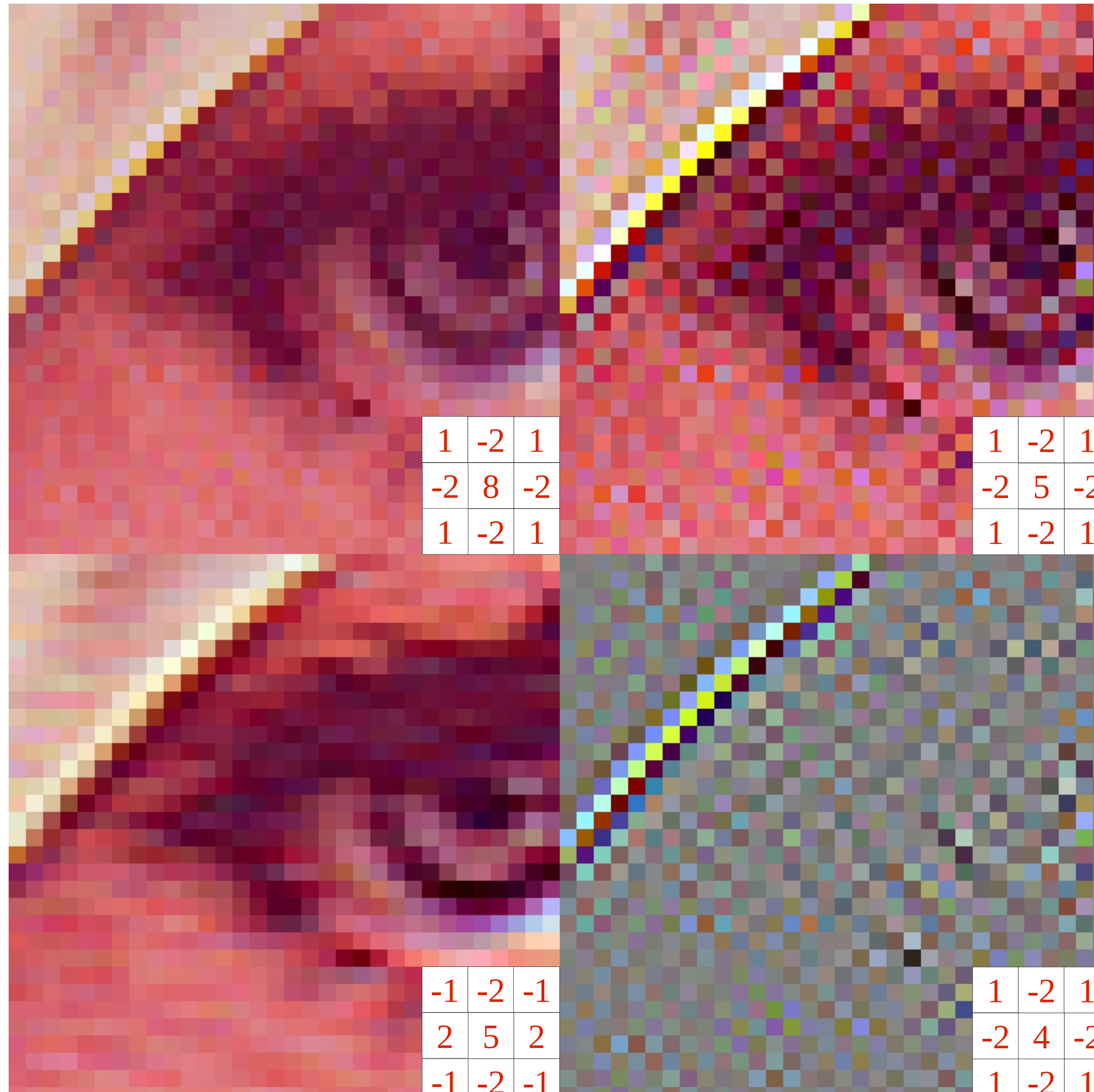
Smoothing filters



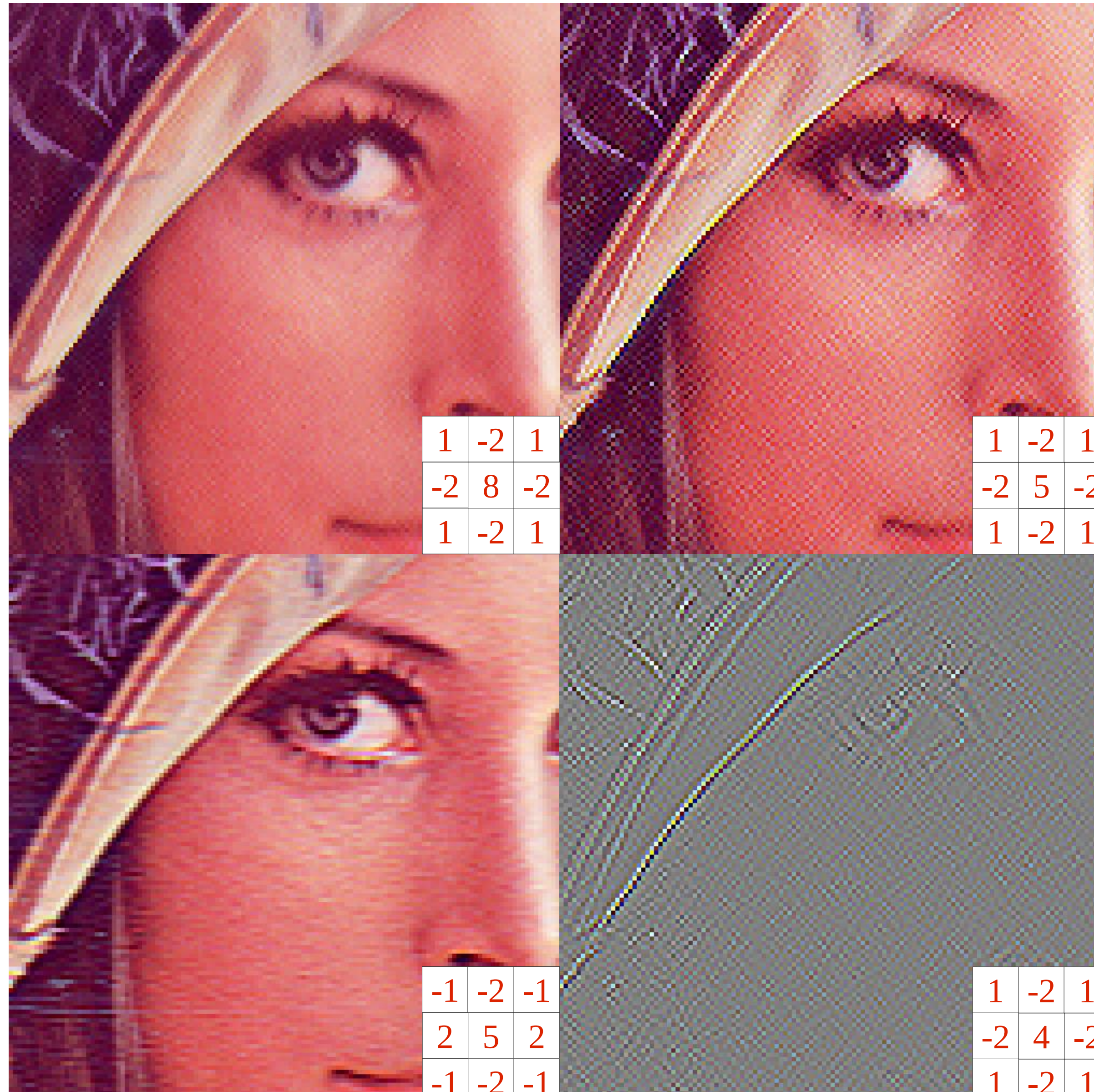


Smoothing filters

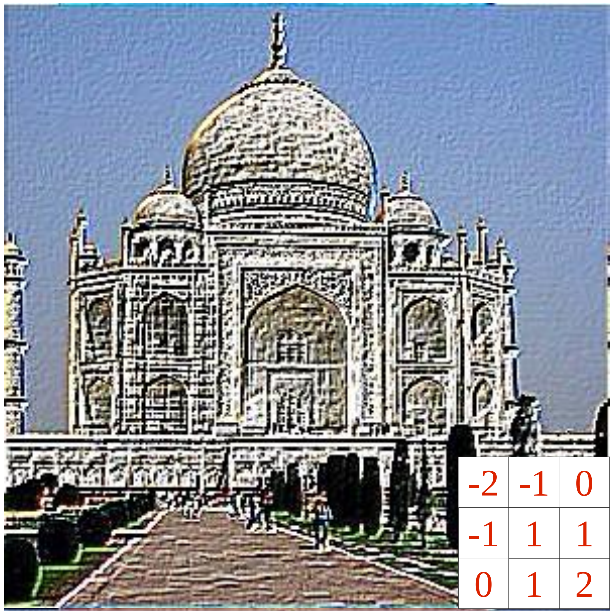
Sharpening filters

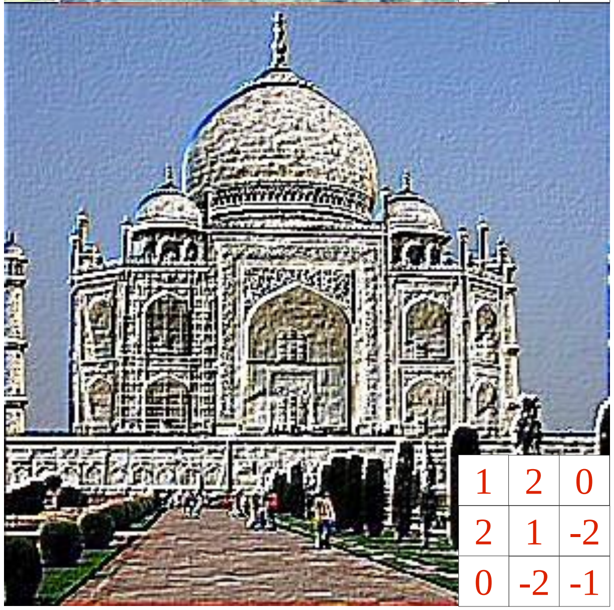


Sharpening filters



Spatial enhancement





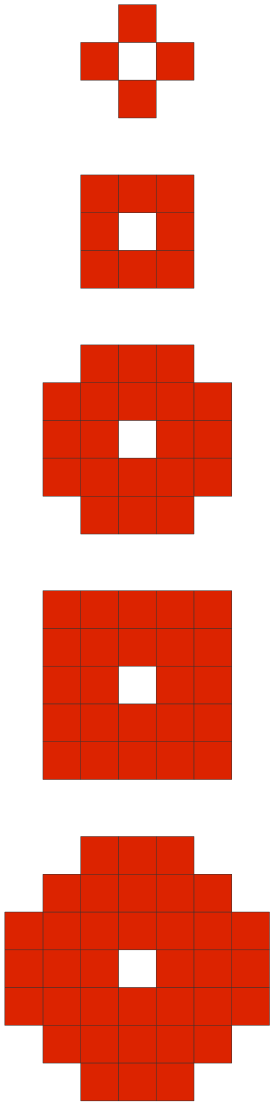
Rank filters

k-th neighbour

Minimum filter (**k=1**)

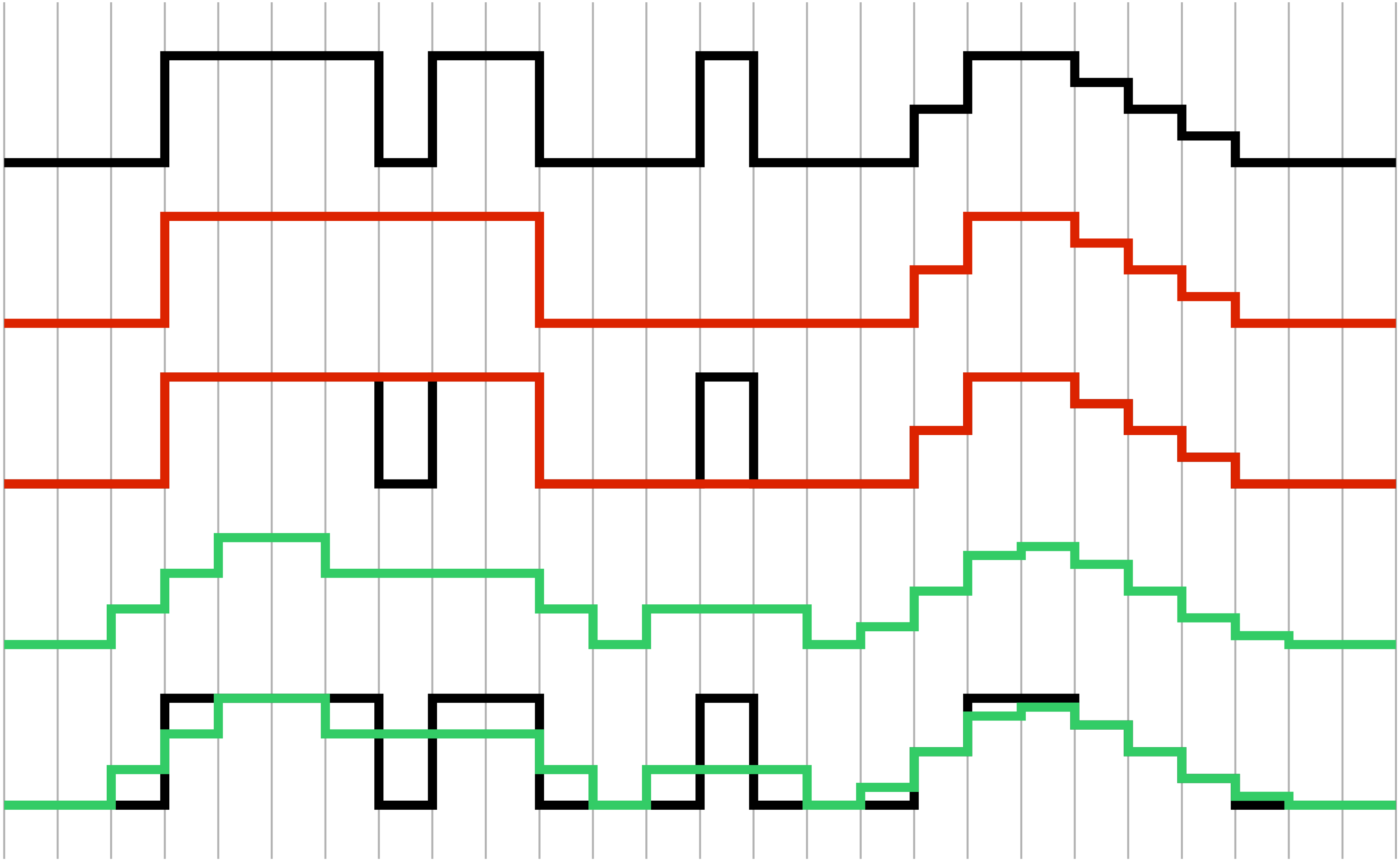
Maximum filter (**k=n**)

Median filter (**k=n/2**)

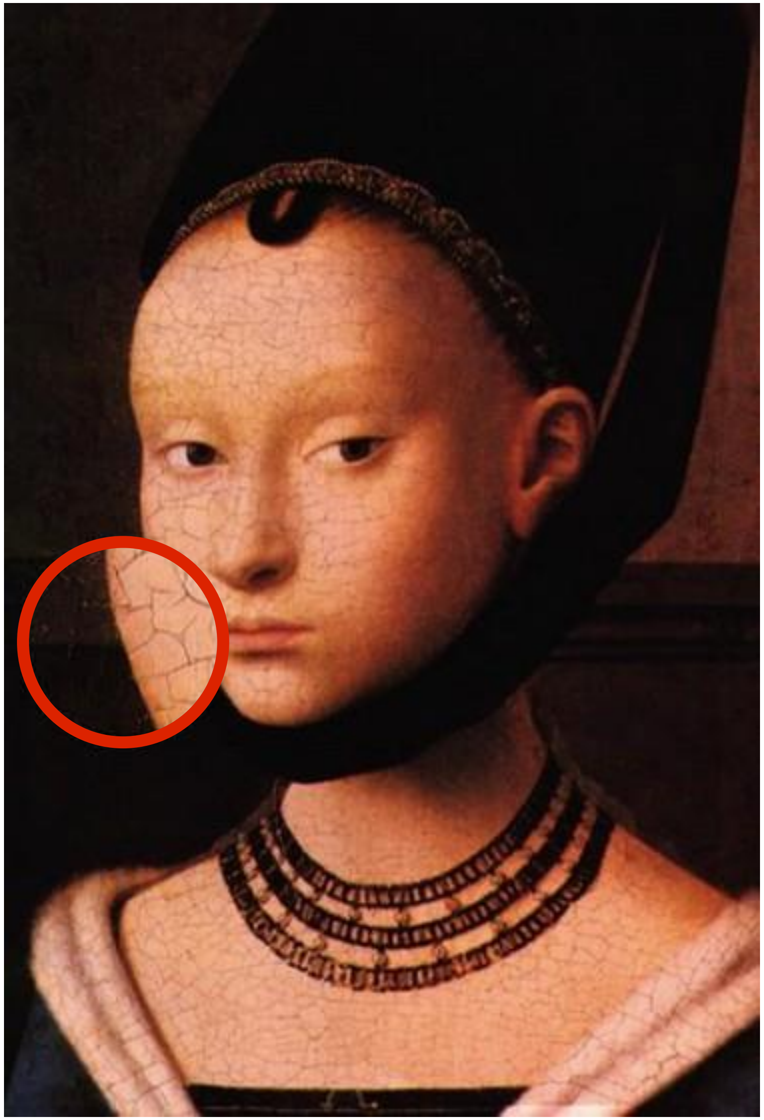


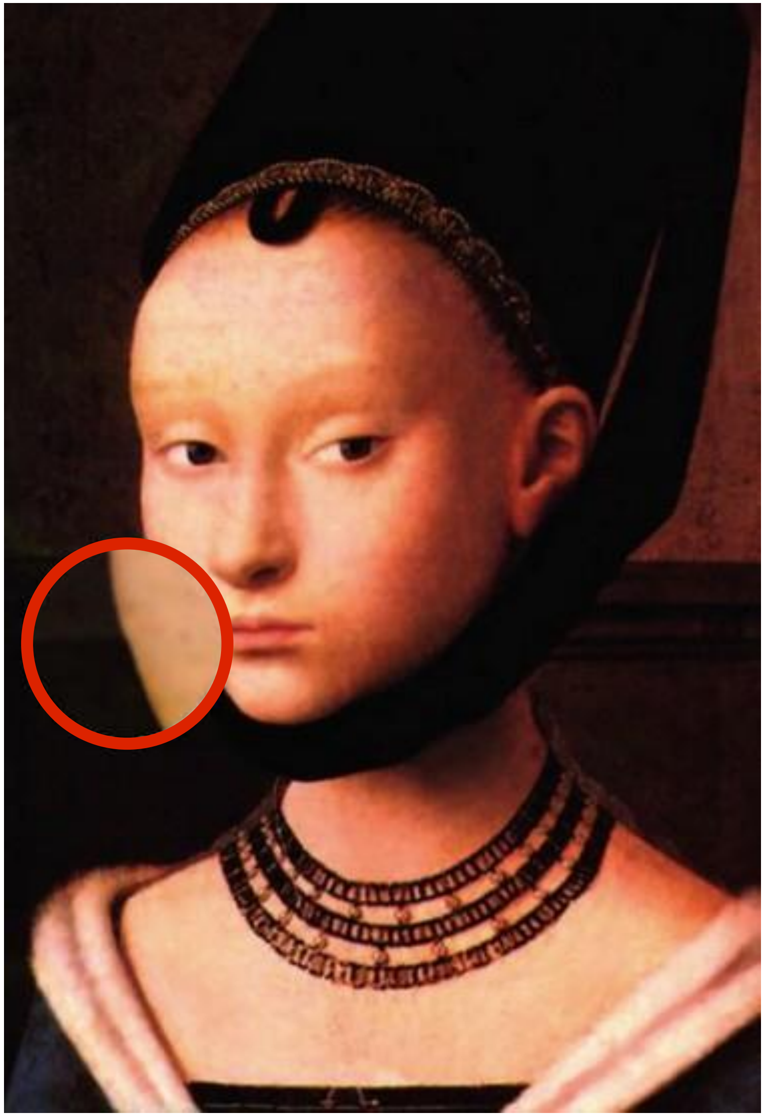
Median

Median / averaging



Median





Linear vs Rank

Diadic decomposition - separable filters

1	2	1

1
2
1

1	2	1
2	4	2
1	2	1

Only with linear filters (not always)

Image mathematics

Image-value v. **image-image**

Addition / subtraction / averaging

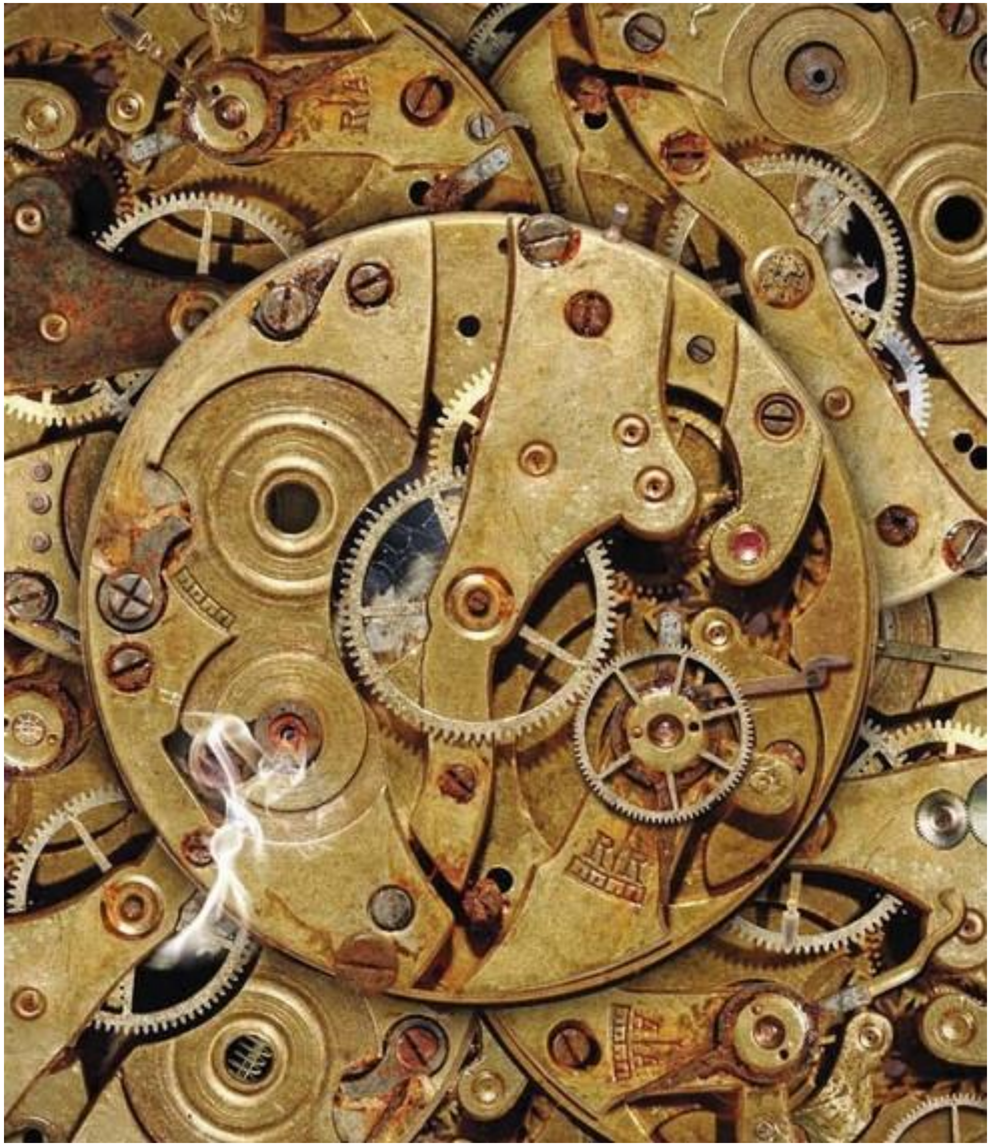
Multiplication / division / normalisation

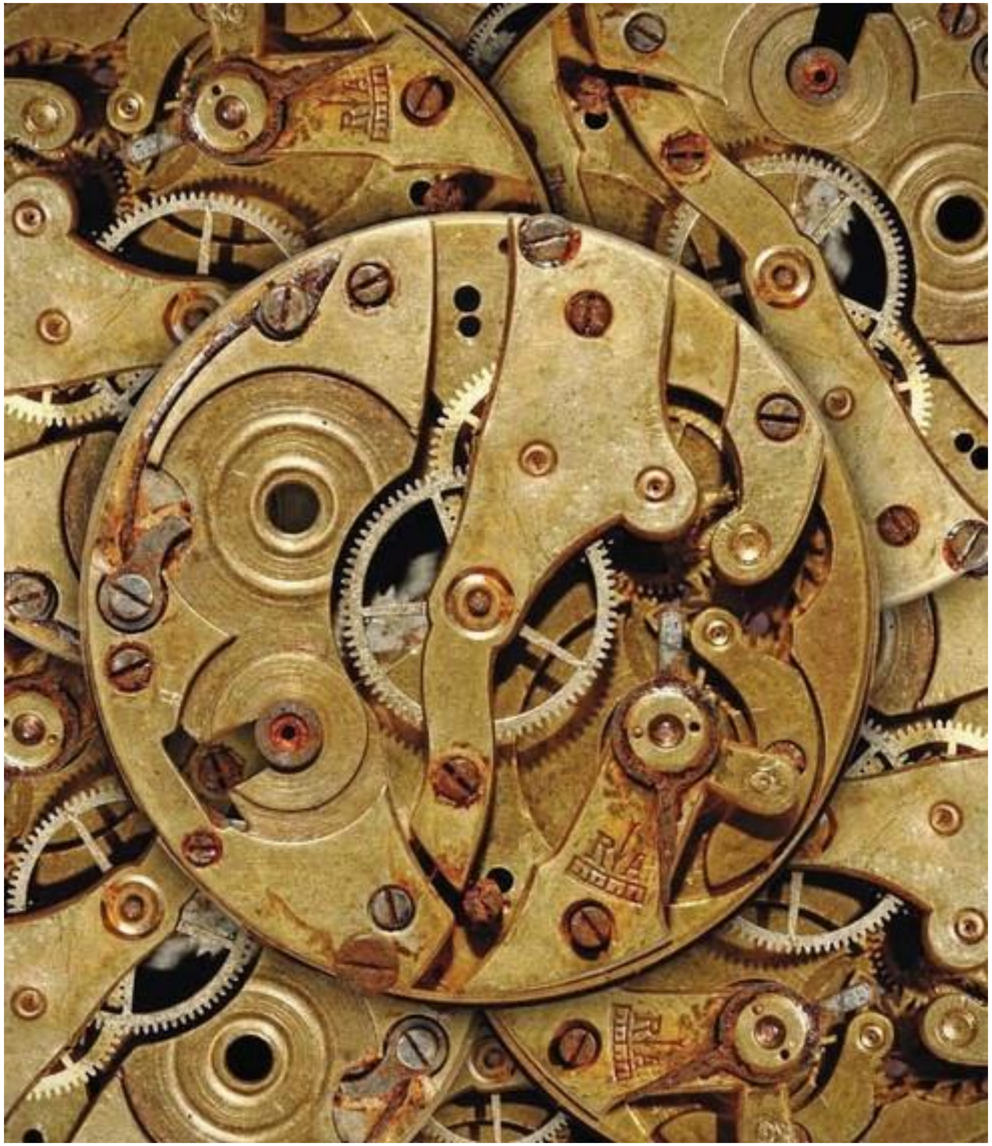
Maximum / minimum

Logical operations (mainly binary images)

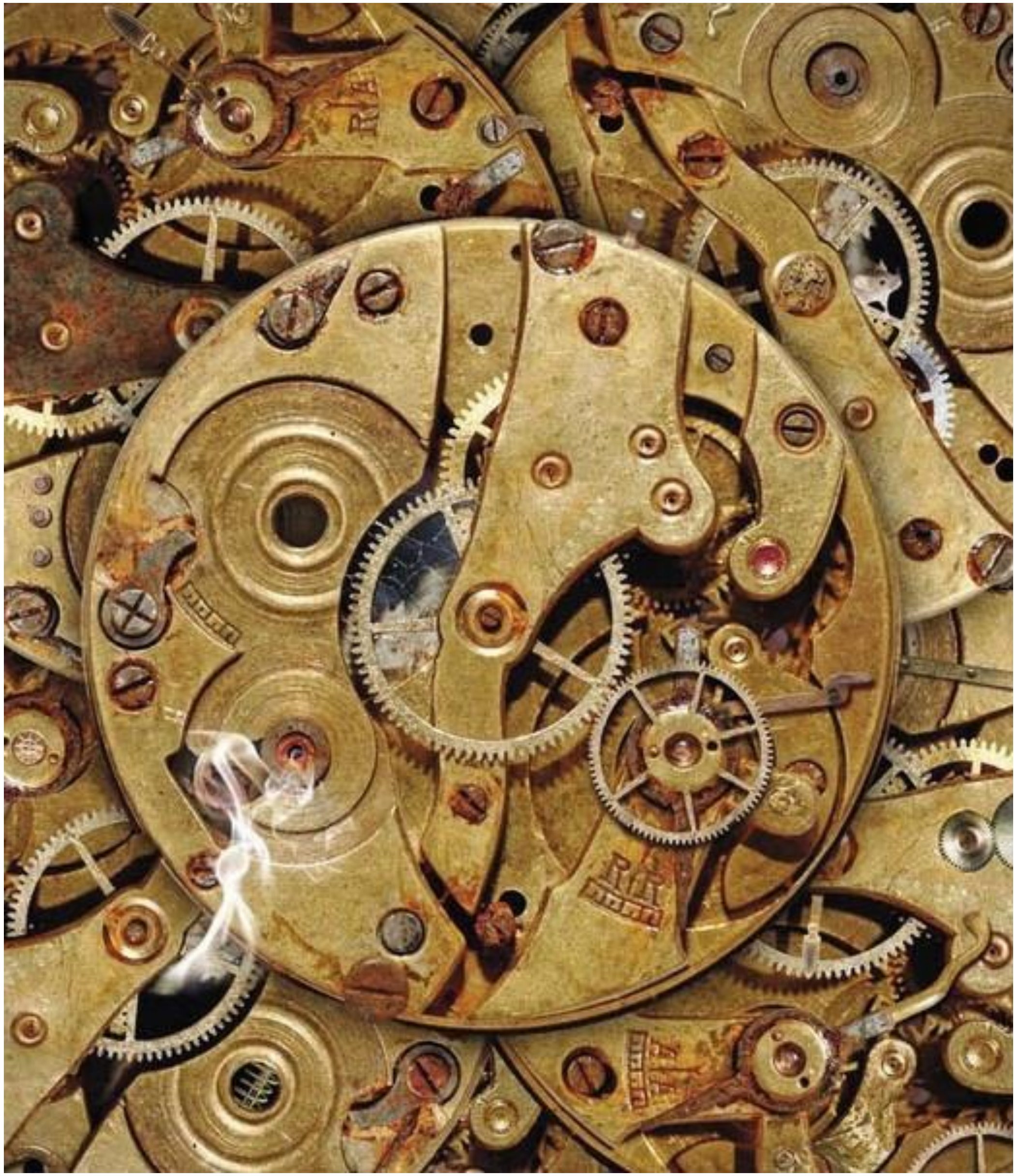
Texturing (multiplication)

Difference

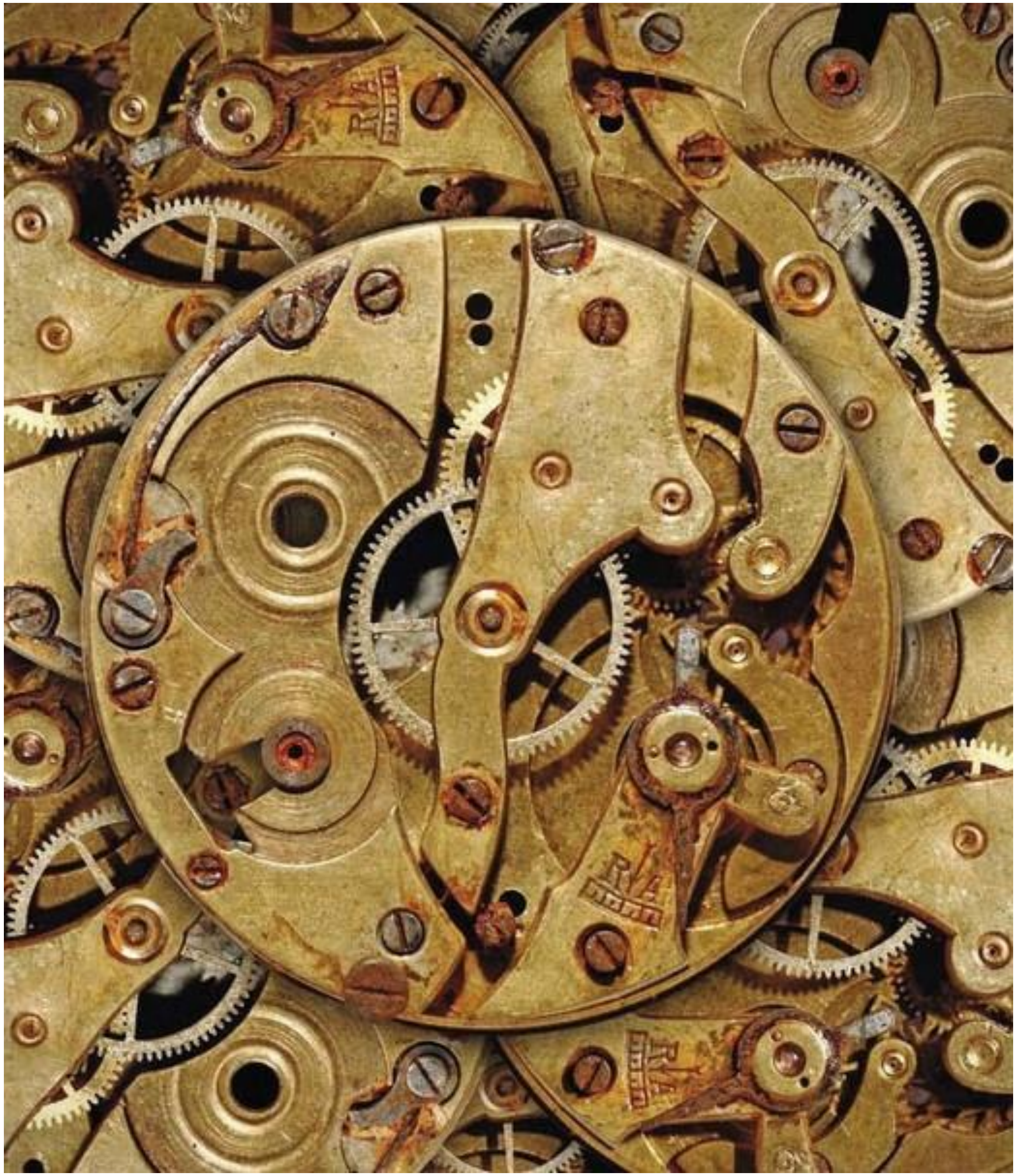




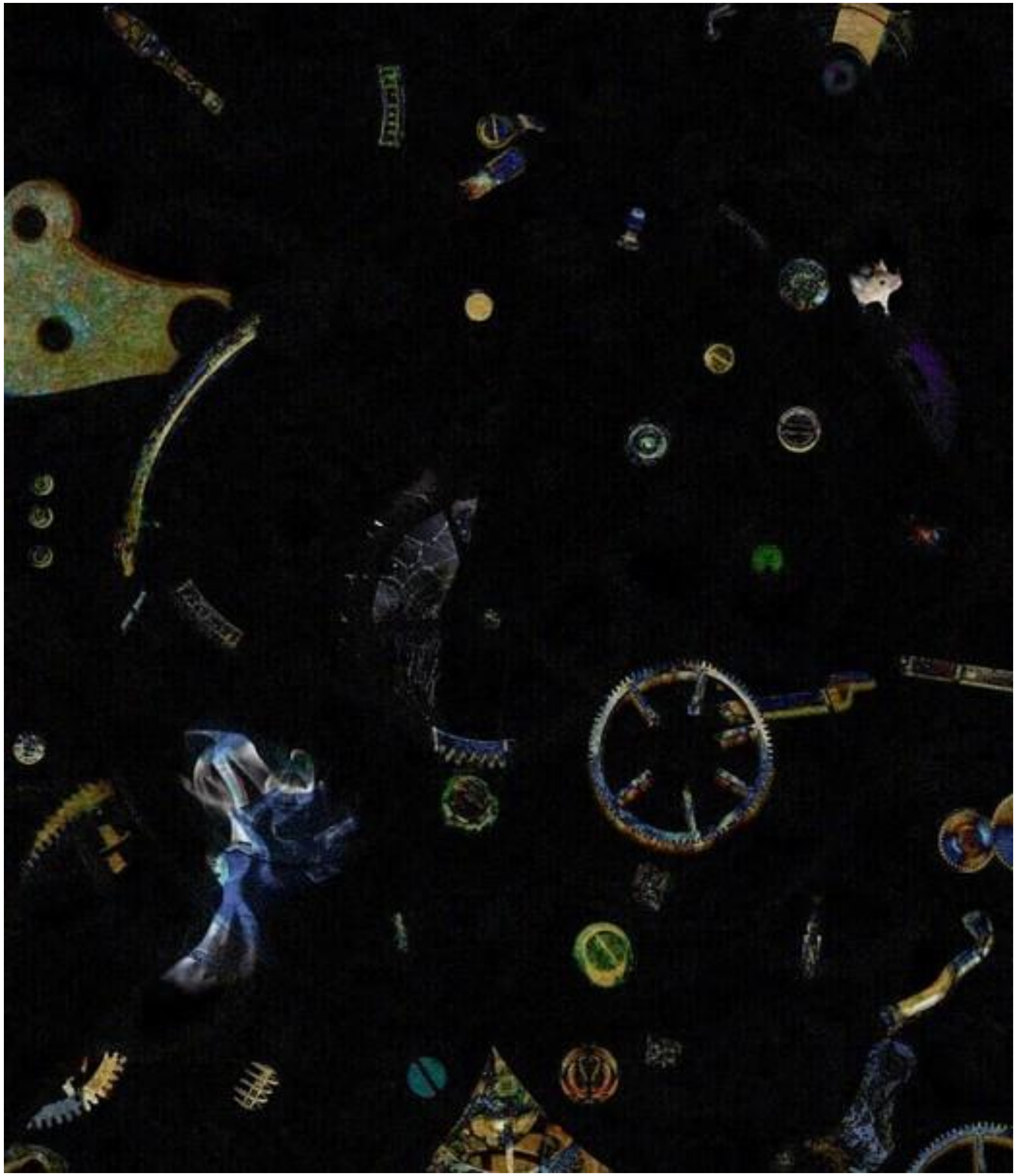
Difference



Difference

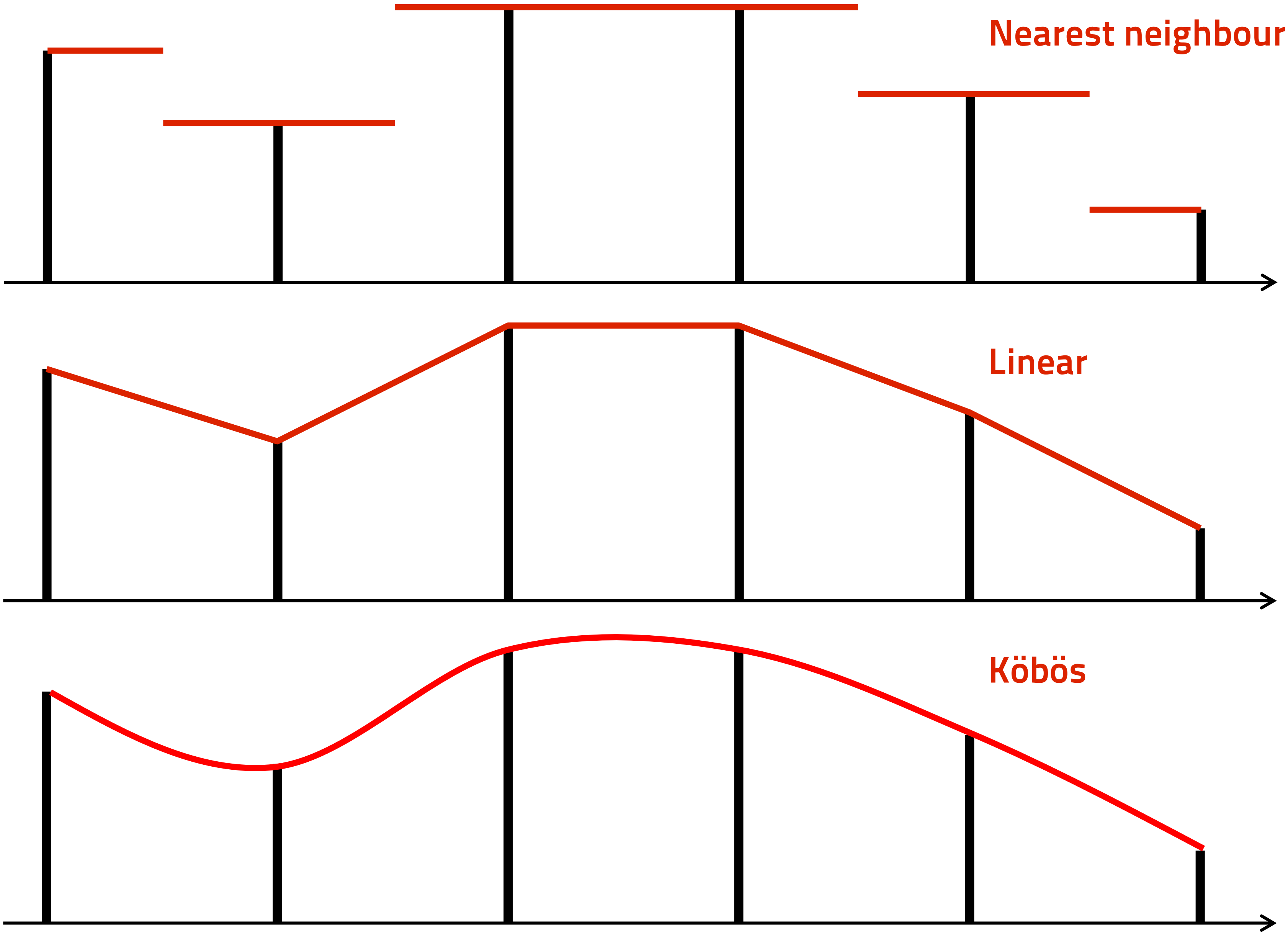


Difference



Background separation

Interpolation

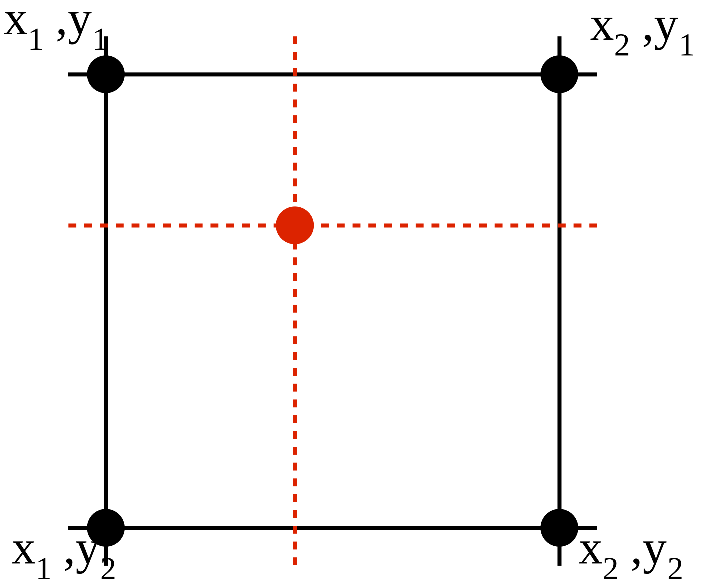


Bilinear interpolation

$$f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} \cdot f(x_1, y_1) + \frac{x - x_1}{x_2 - x_1} \cdot f(x_2, y_1)$$

$$f(x, y_2) = \frac{x_2 - x}{x_2 - x_1} \cdot f(x_1, y_2) + \frac{x - x_1}{x_2 - x_1} \cdot f(x_2, y_2)$$

$$f(x, y) = \frac{y_2 - y}{y_2 - y_1} \cdot f(x, y_1) + \frac{y - y_1}{y_2 - y_1} \cdot f(x, y_2)$$



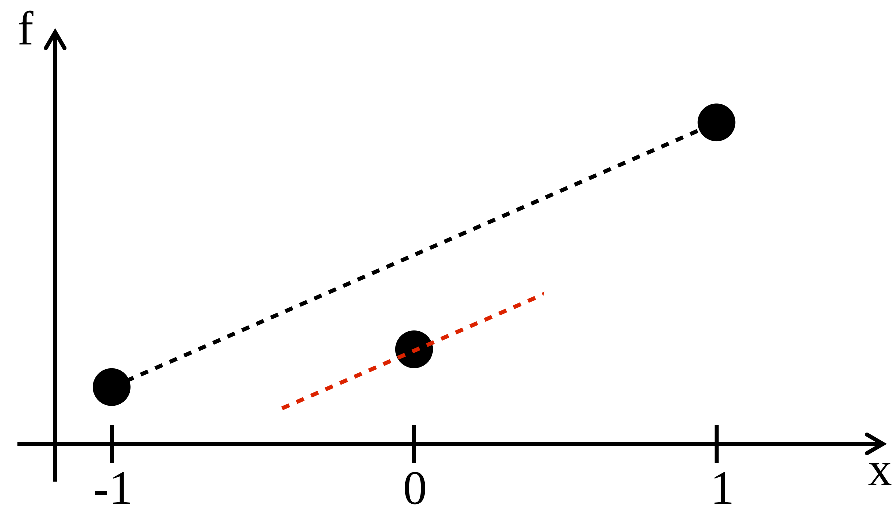
Bicubic Interpolation

$$f(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}\, x^i y^j$$

$$\frac{\partial f(x,y)}{\partial x} = \sum_{i=1}^{3}\sum_{j=0}^{3} a_{ij}\, i x^{i-1} y^j$$

$$\frac{\partial f(x,y)}{\partial y} = \sum_{i=0}^{3}\sum_{j=1}^{3} a_{ij}\, x^i j y^{j-1}$$

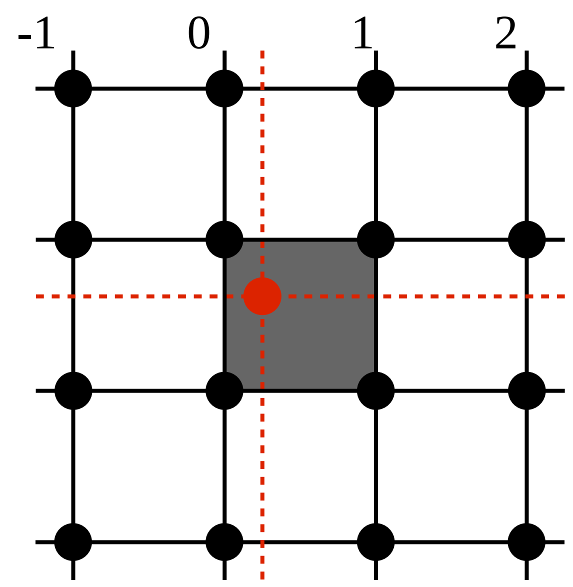
$$\frac{\partial^2 f(x,y)}{\partial x \partial y} = \sum_{i=1}^{3}\sum_{j=1}^{3} a_{ij}\, i x^{i-1} j y^{j-1}$$



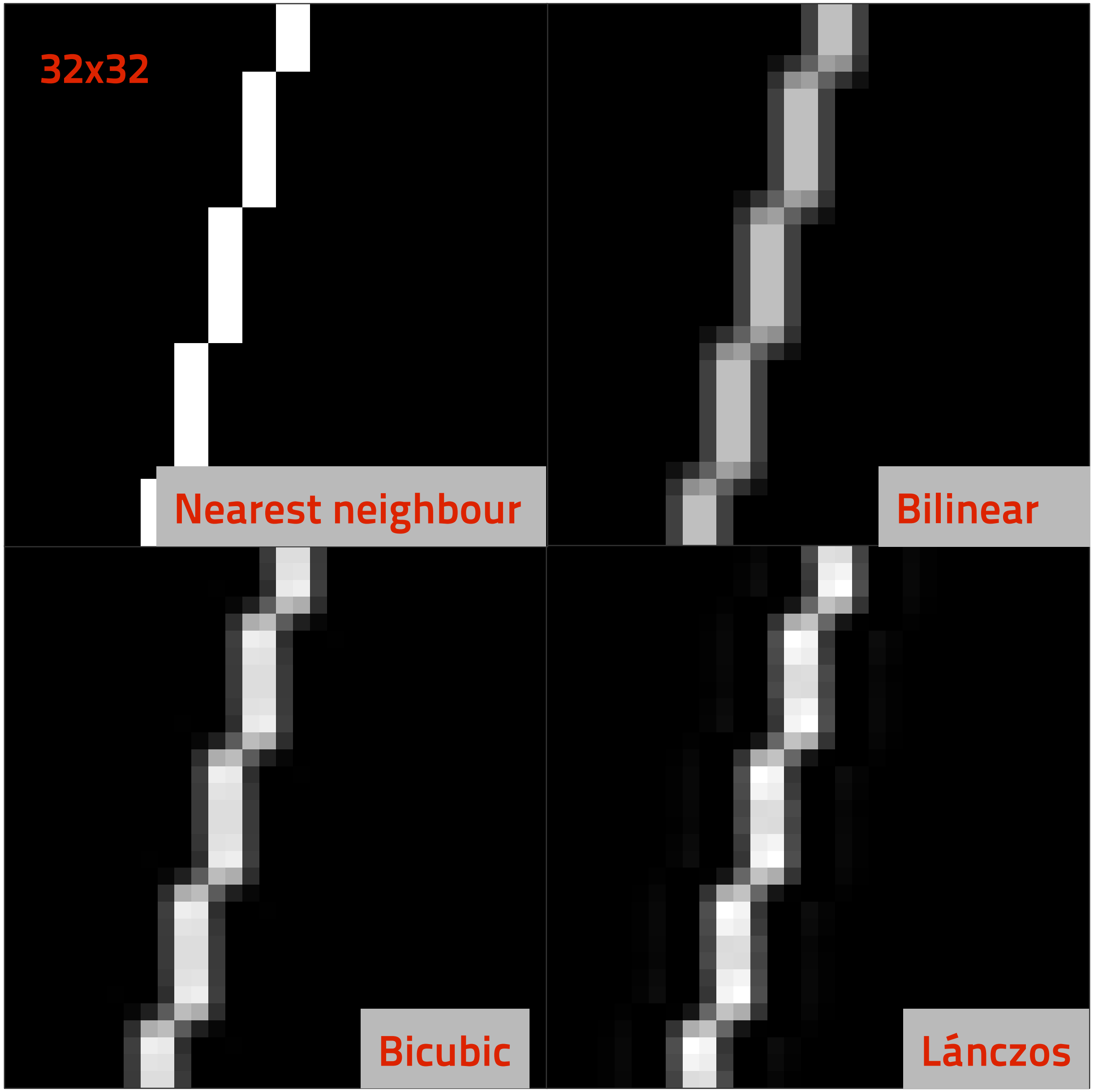
$$\frac{\partial f(x,y)}{\partial x} = \frac{f(x+1,y) - f(x-1,y)}{2}$$

$$\frac{\partial f(x,y)}{\partial y} = \frac{f(x,y+1) - f(x,y-1)}{2}$$

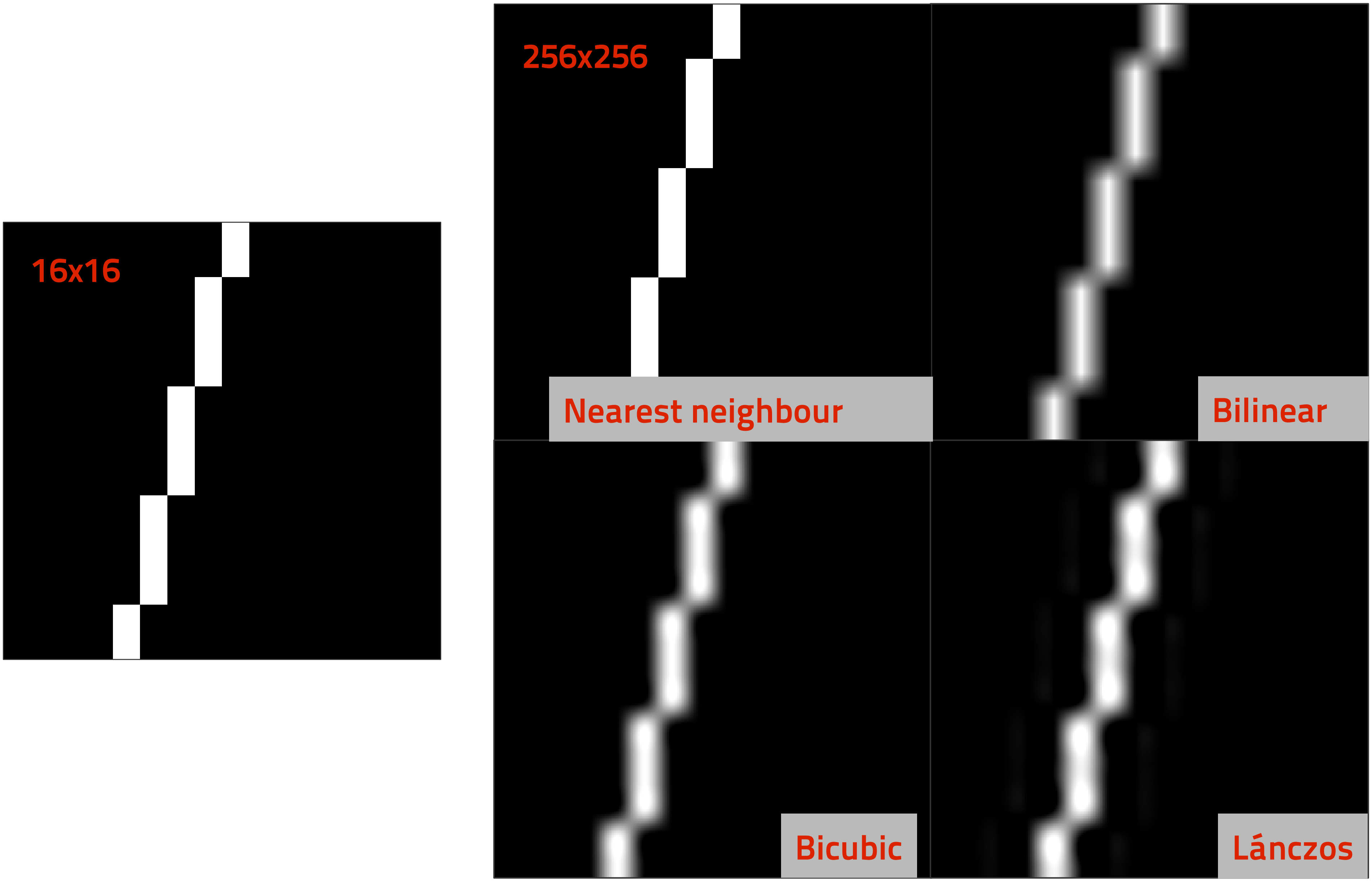
$$\frac{\partial^2 f(x,y)}{\partial x \partial y} = \frac{f(x+1,y+1) - f(x-1,y) - f(x,y-1) + f(x,y)}{4}$$



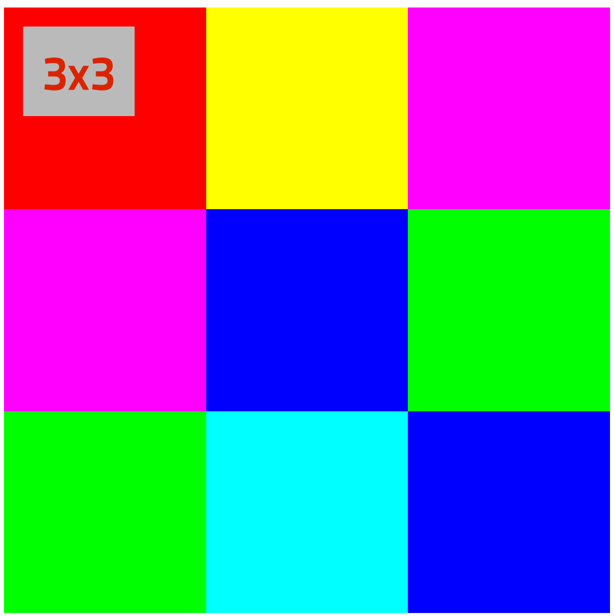
Interpolation

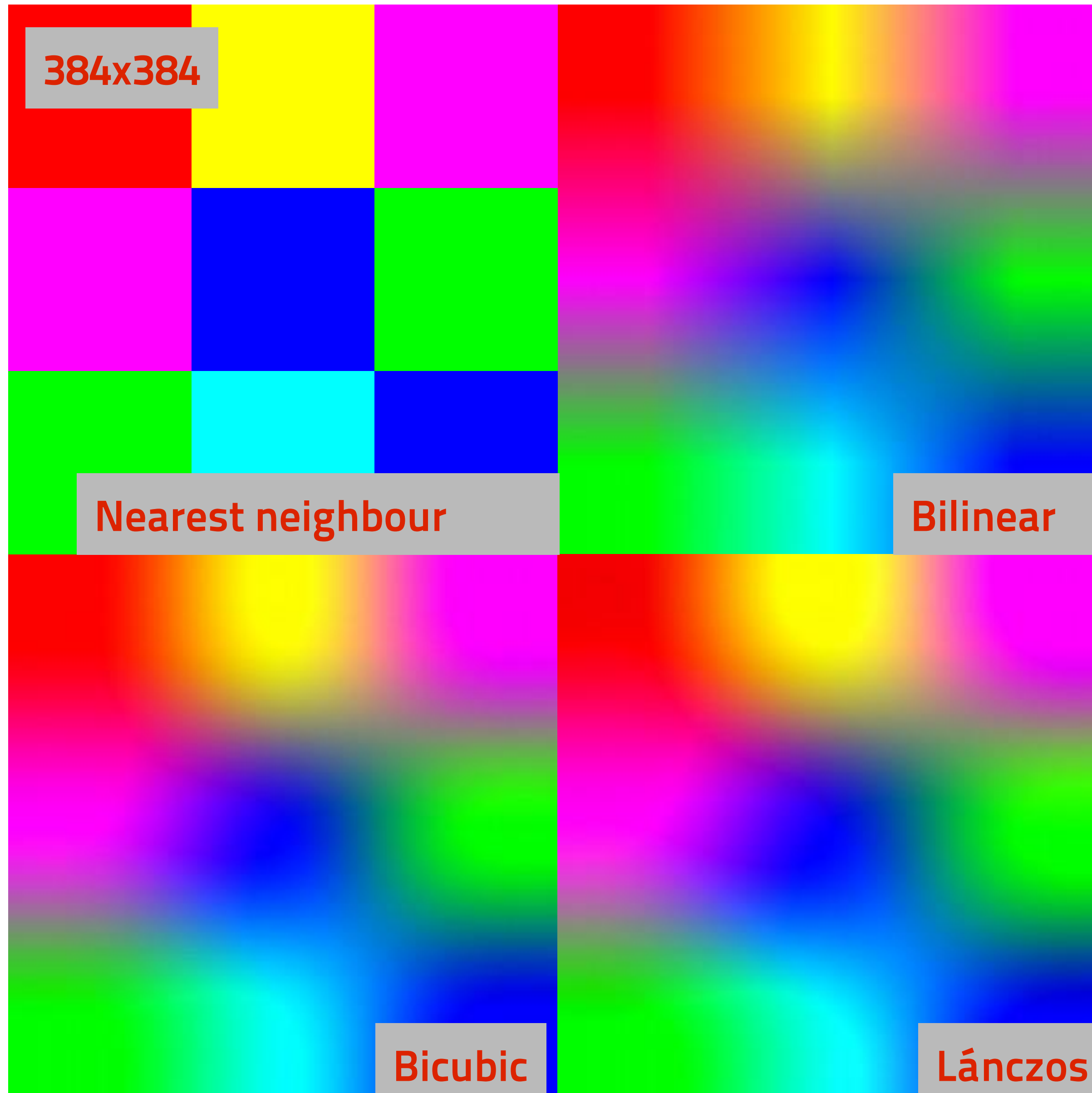


Interpolation



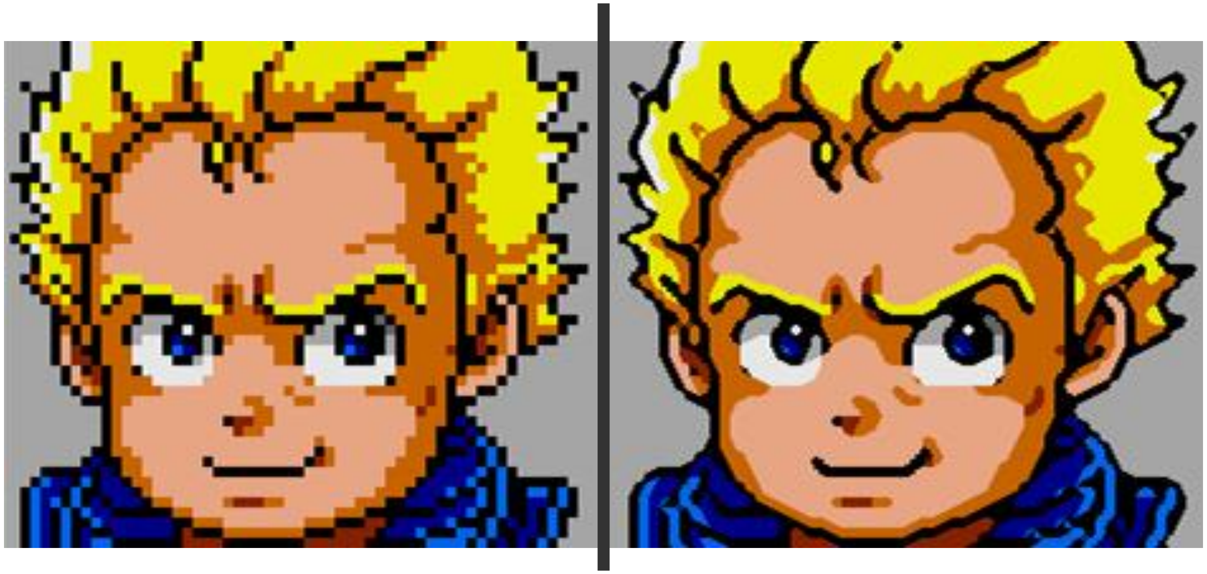
Interpolation

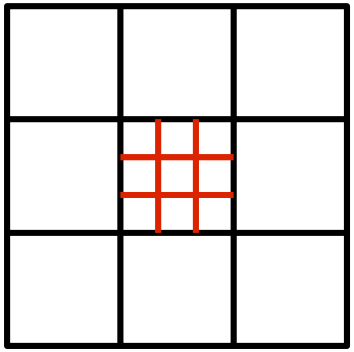




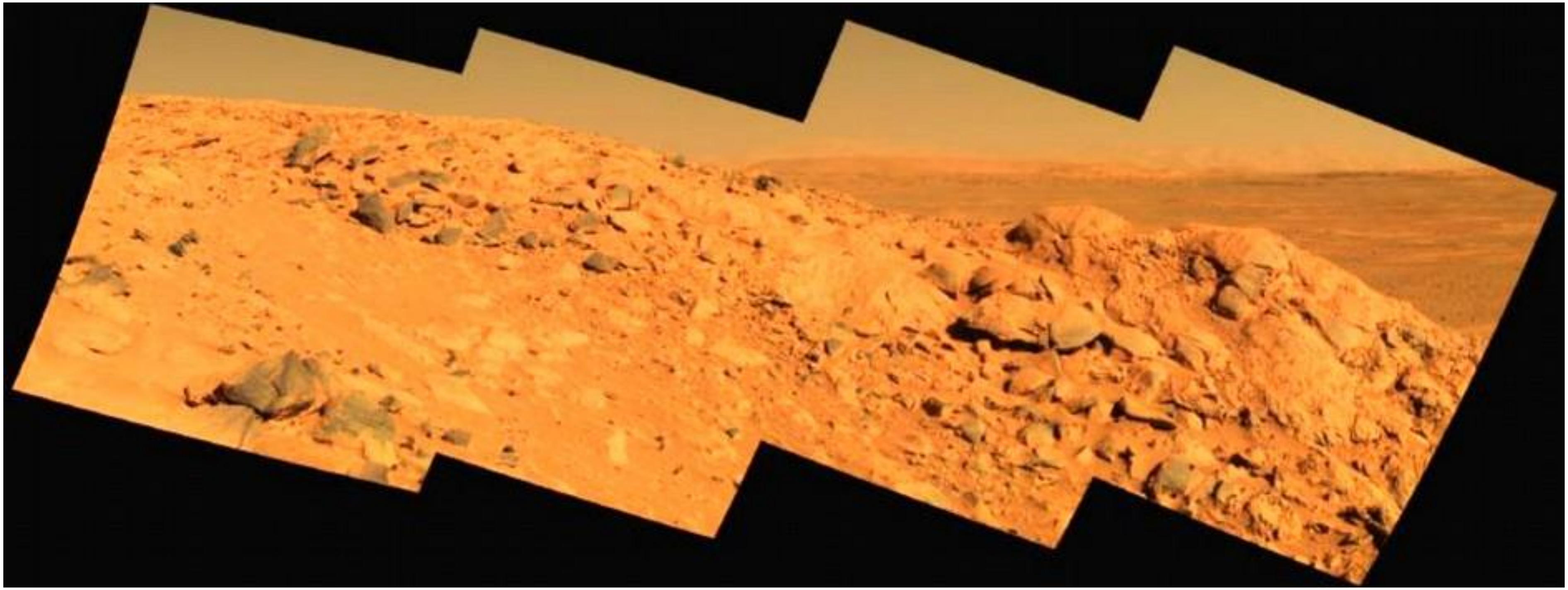
Complex interpolation

Scale2X, 3X, 4X

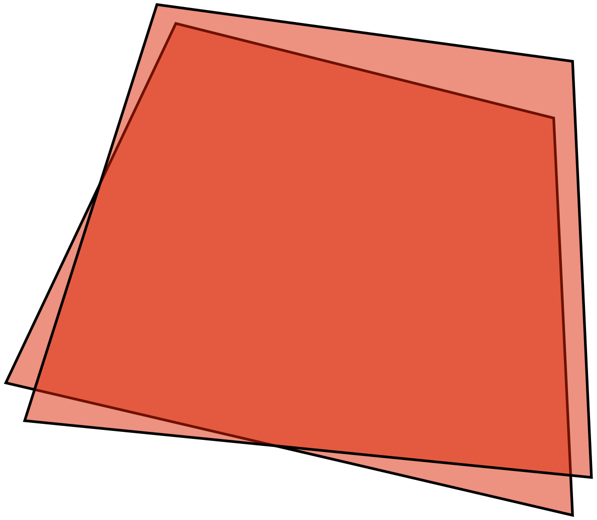




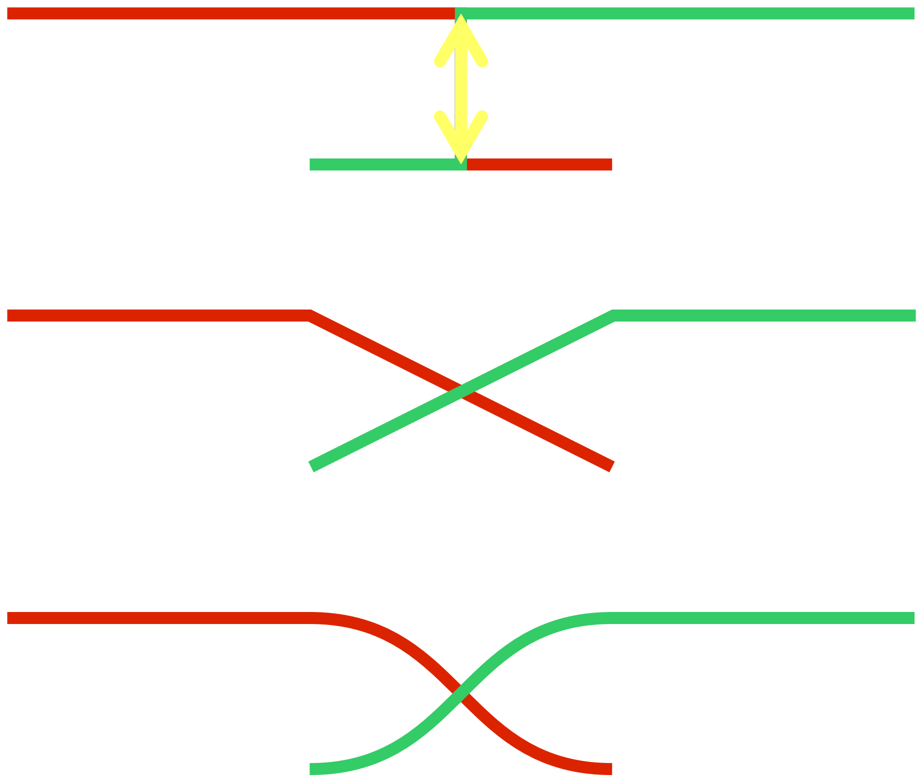
Registration



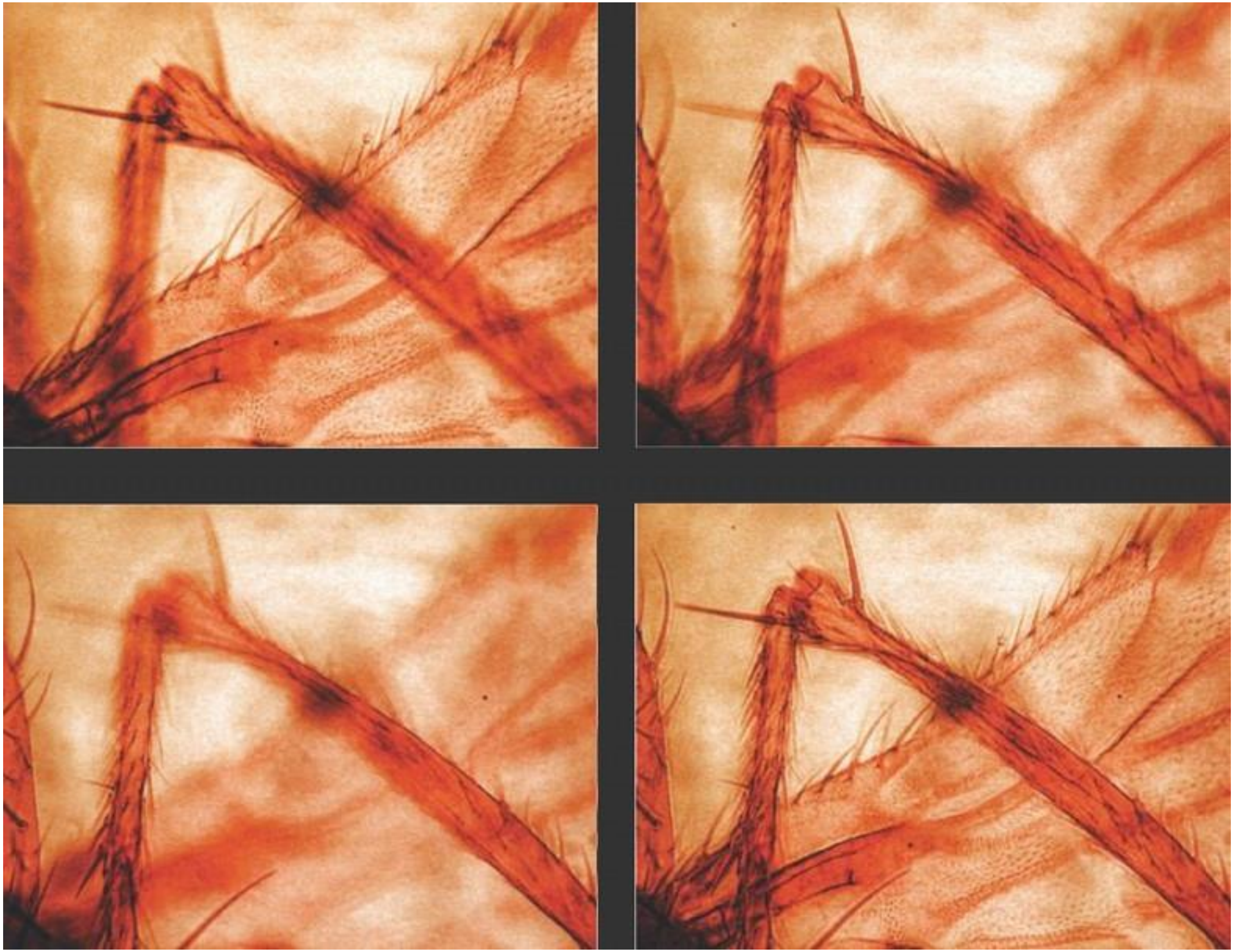
Overlap



Overlap



Combination - focus



Combination - brightness