Image preprocessing in the spatial domain, local neighborhood operations

Václav Hlaváč

Czech Technical University in Prague
Czech Institute of Informatics, Robotics and Cybernetics
160 00 Prague 6, Jugoslávských partyzánů 1580/3, Czech Republic
http://people.ciirc.cvut.cz/hlavac, vaclav.hlavac@cvut.cz
also Center for Machine Perception, http://cmp.felk.cvut.cz

Thanks to Tomáš Svoboda for several slides.

Outline of the lecture:

- Noise filtration (no edge detection here).
- Noise and its statistical nature.
- Space invariant filters.
- Discrete 2D convolution.
- Separable filters.
- Nonlinear noise filtration.

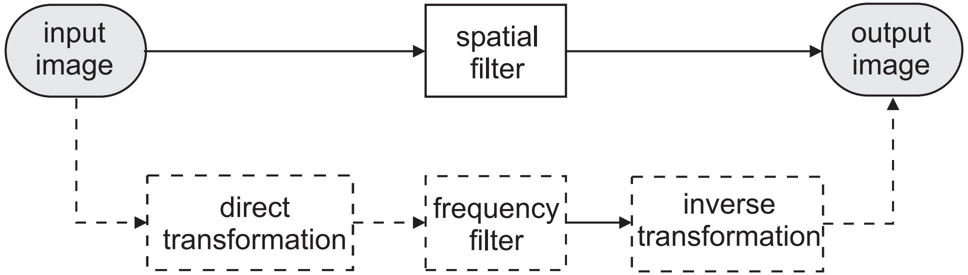
Image preprocessing (filtering), the introduction

The input is an image, the output is an image too.

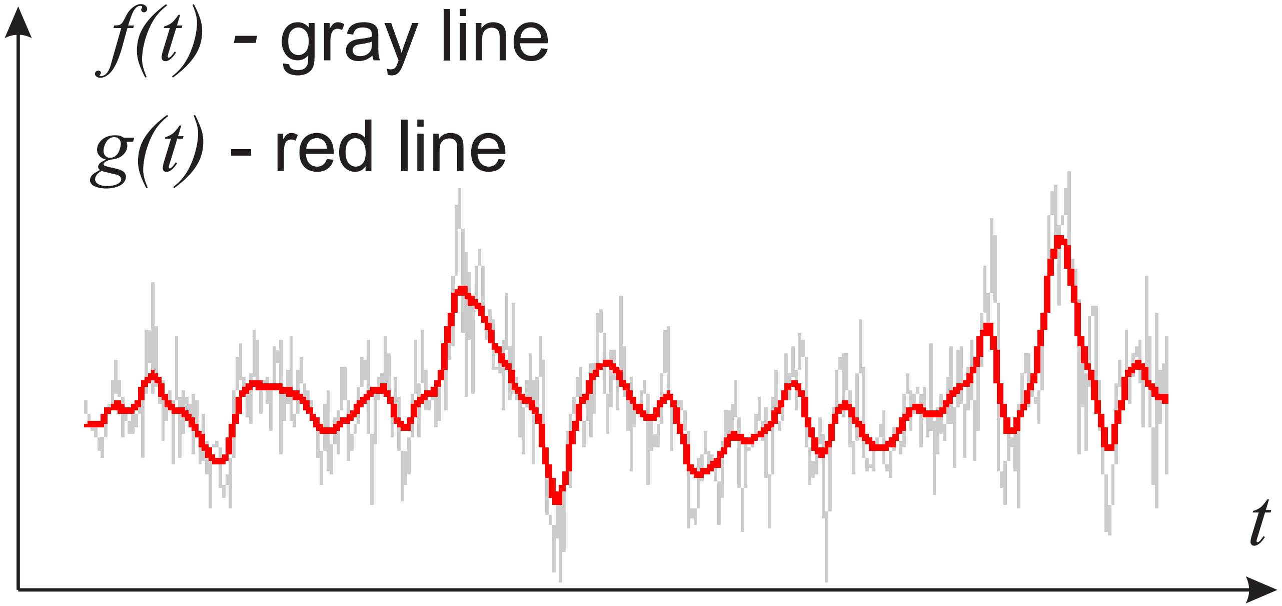
The image is not interpreted. The image content is not taken into account.

The aim

- To suppress the **distortion** (e.g., correction of the geometric distortion caused by the Earth spherical shape of an satellite image).
- **Contrast** enhancement (useful only if the human looks at the image).
- Filtering removes unwanted components, e.g. suppresses noise.
- **Enhancement of some image features** needed for further image processing, e.g. edge elements finding.

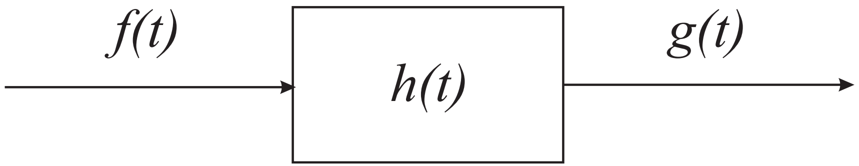


1D signal filtering, motivating example



1D signal linear filtering, convolution

- Consider a 1D filter h (also system, plant) depending on time t, which is linear and time invariant.
- Consider continuous signals, the input signal $f(t)$ and the output signal $g(t)$.

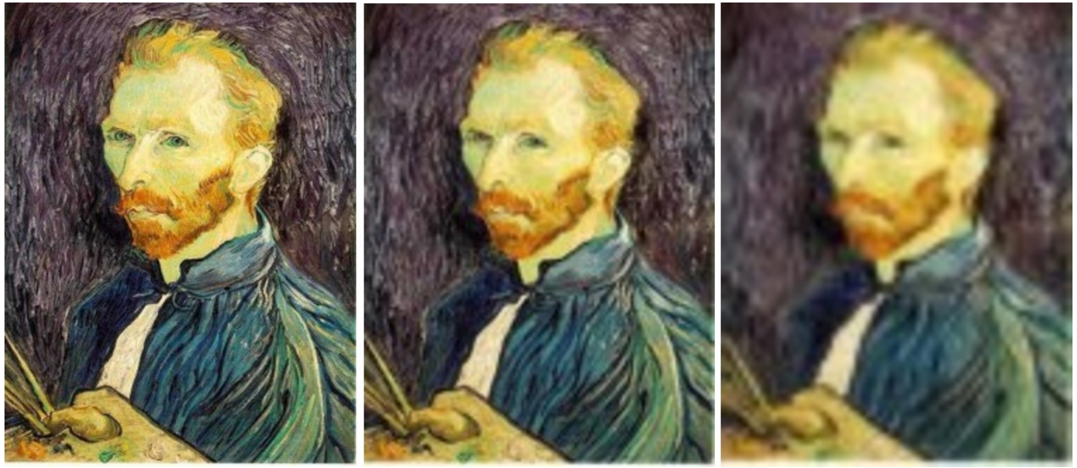


- The output of the filter h can be calculated using the convolution integral

$$g(t) = f(t) * h(t) = \int_{-\infty}^{\infty} f(t)\, h(t-\tau)\, \mathrm{d}\tau = \int_{-\infty}^{\infty} f(t-\tau)\, h(t)\, \mathrm{d}\tau \,.$$

2D image filtering, motivating example

smoothing



sharpening

Image filtering × image warping

Image filtering changes the range of the image function.

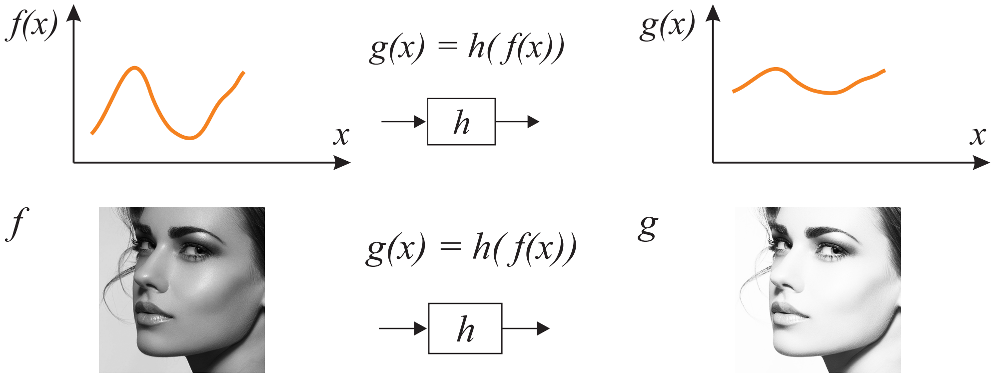
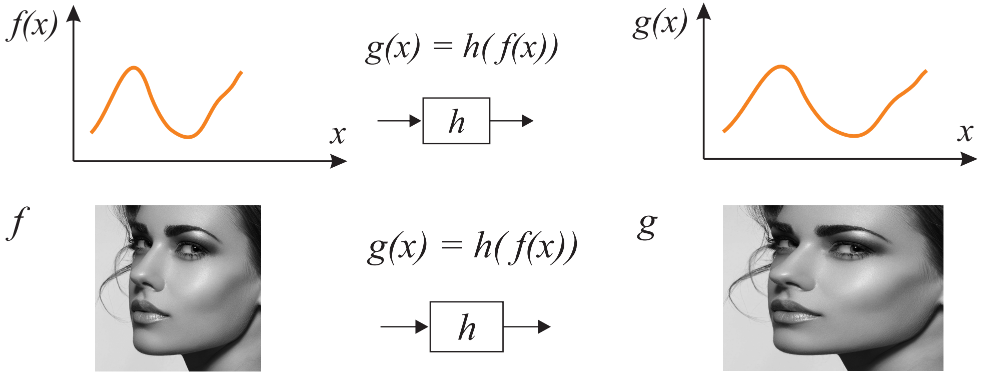


Image warping changes the domain of the image function.



Local image preprocessing operations

- The specific knowledge about the image and distortions (i.e. semantics) in not used in preprocessing.
- Two possible taxonomies of local image preprocessing operations:
  - Taxonomy from the usage point of view:
    1. Smoothing (noise filtration).
    2. Edge detection (gradient operators, image sharpening).
  - Taxonomy according to the character of the mathematical description:
    1. Linear.
    2. Nonlinear.

Noise in the image

- (Image) noise is a random variation of the (image) function, which is mostly unwanted.
- In the case of 2D image function $f(x, y)$, the random variation can be in the value of f as well as in the position rendomness of observed samples (x, y). The latter case is considered less often.
- Noise is most often modeled mathematically using probabilistic tools, e.g. by the noise probability distribution.
- Sources of the image noise are, e.g.: illumination fluctuation, light-sensitive sensor noise, quantization effects, image compression artifacts, and a finite computation precision.

Noise

- Let $f(x,y) = I(x,y)$ be the image; $N(x,y)$ be noise.
- The additive noise is abstracted by $I + N$.
- The multiplicative noise is given as $I(1 + N)$.

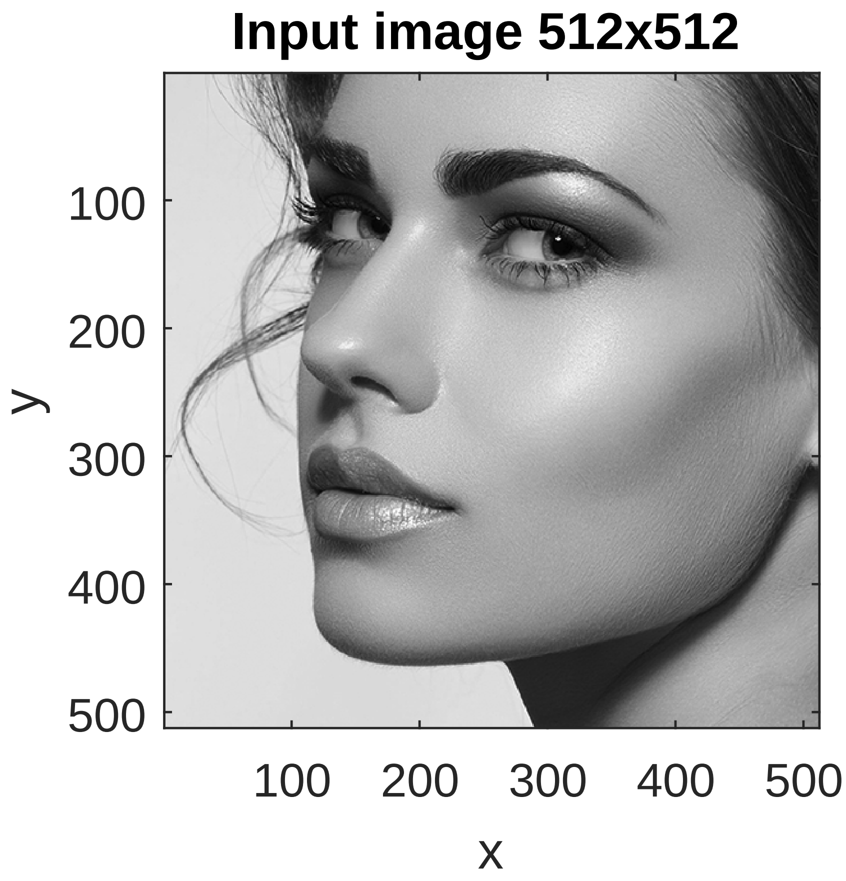
Noise independent of the signal

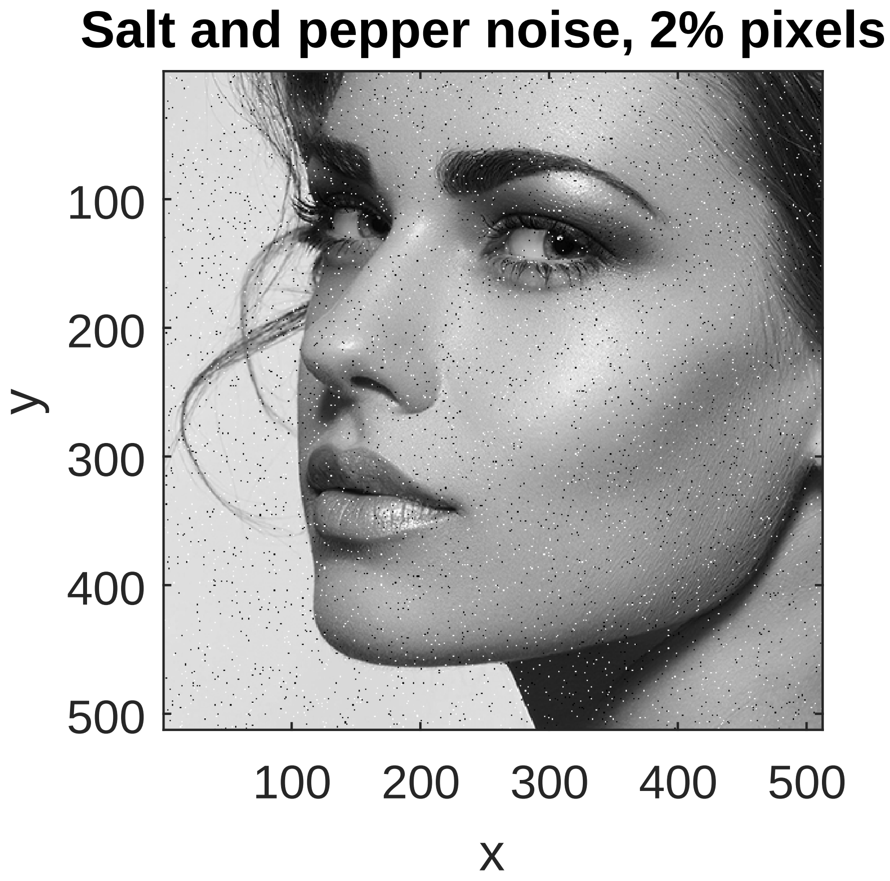
- Salt and pepper noise, also impulse or binary noise: contains random occurrences of black and white pixels.
- Gaussian noise: variations in intensity drawn from a Gaussian normal distribution.

Noise dependent of the signal

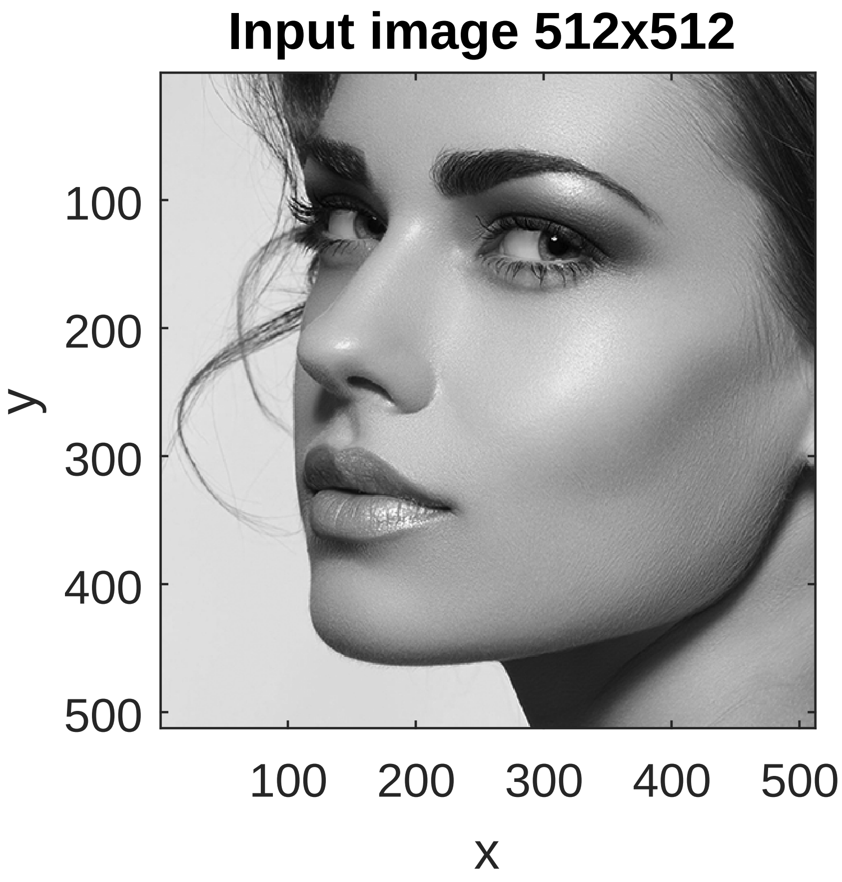
- Speckle noise can be modeled by random values multiplied by pixel values.
- Poisson noise assumes that each pixel of the image $f(x,y)$ is drawn from Poisson distribution of the parameter $\lambda = f_0(x,y)$, where $f_0(x,y)$ is the noise-free image to recover. $P(f(x,y) = k) = \frac{\lambda^k}{k!} e^{-\lambda}$.

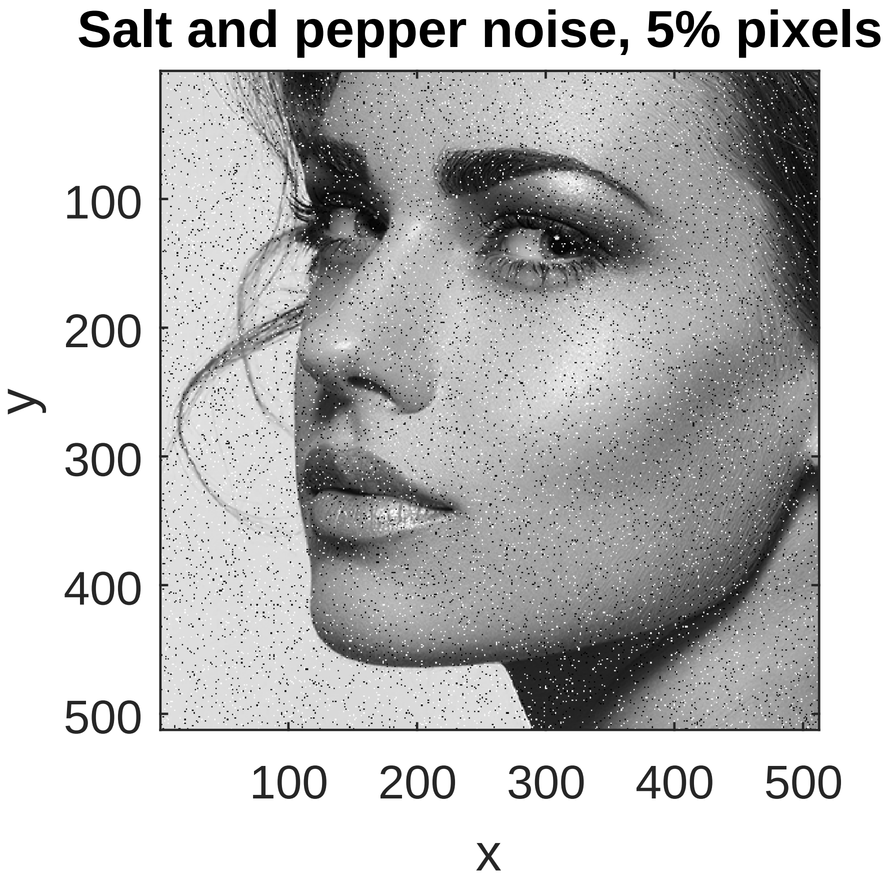
Example, salt & pepper noise, 2% pixels



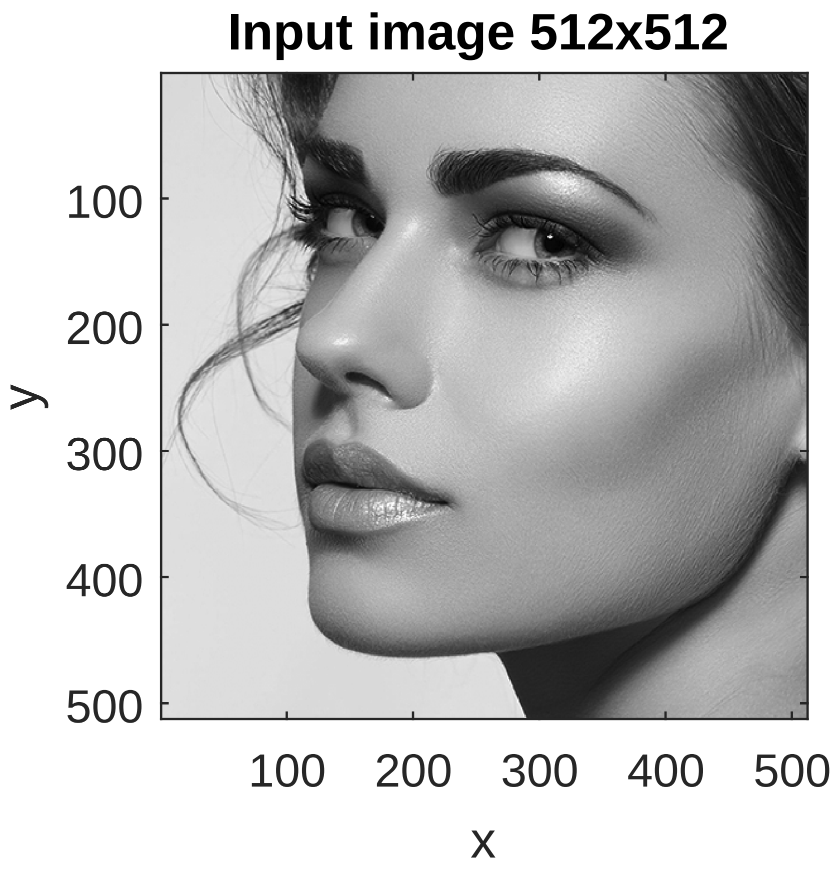


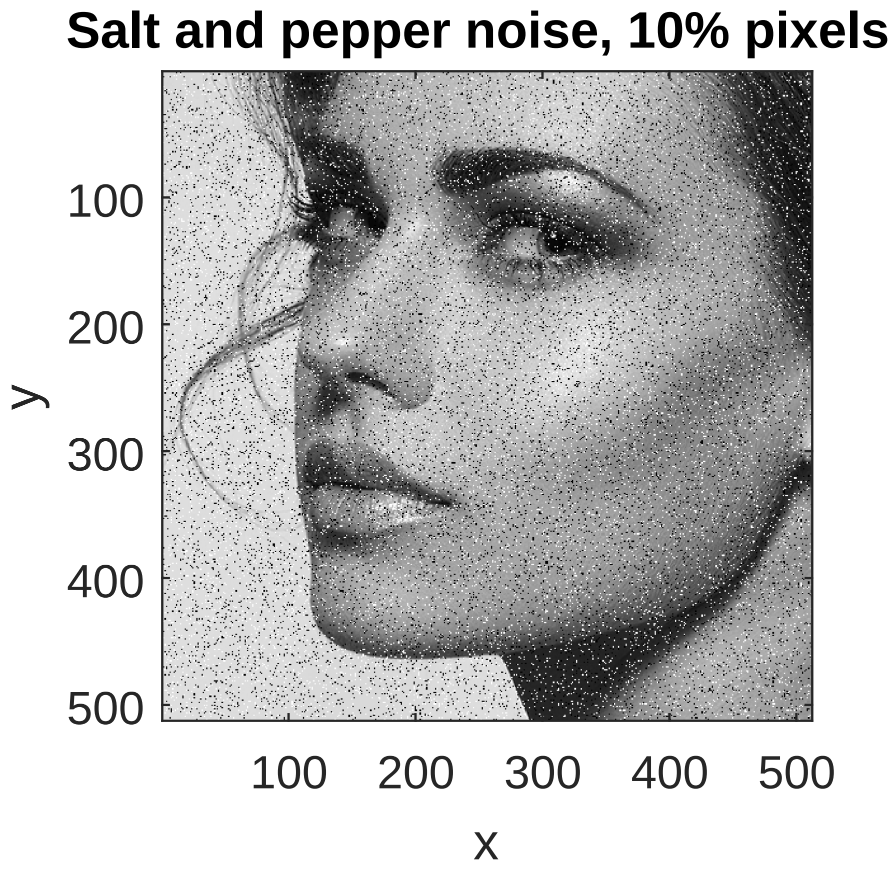
Example, salt & pepper noise, 5% pixels



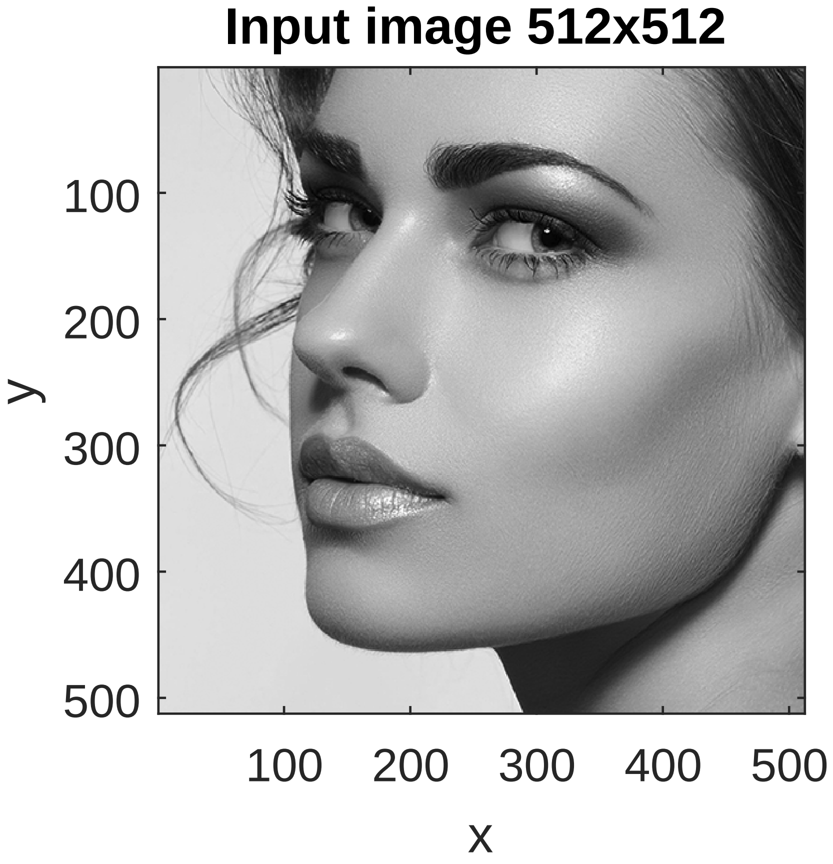


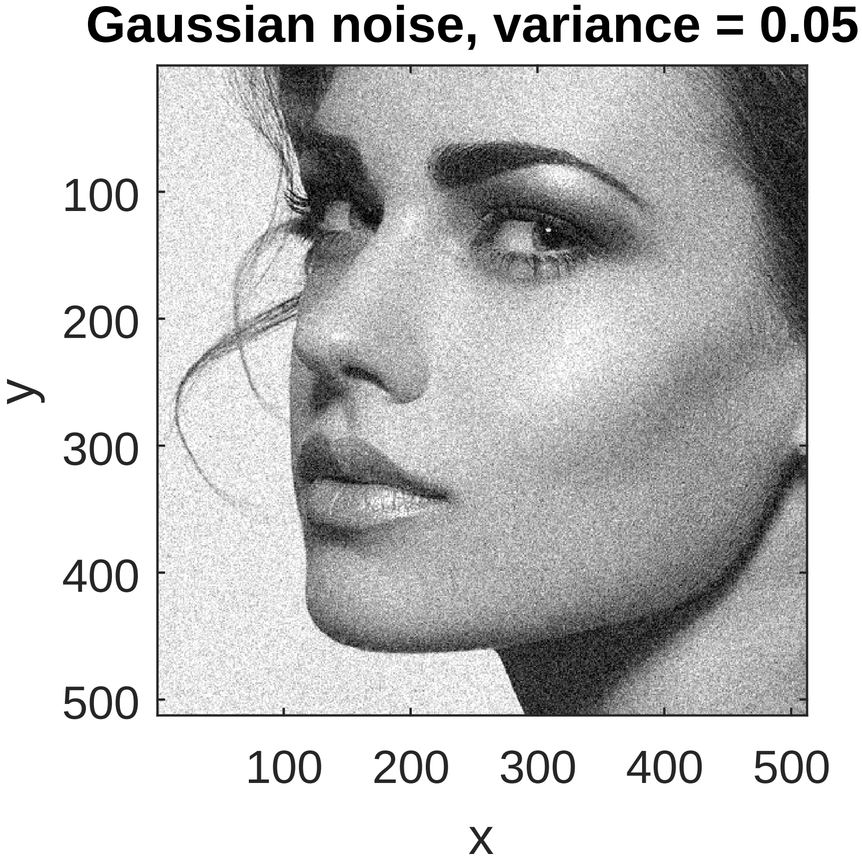
Example, salt & pepper noise, 10% pixels



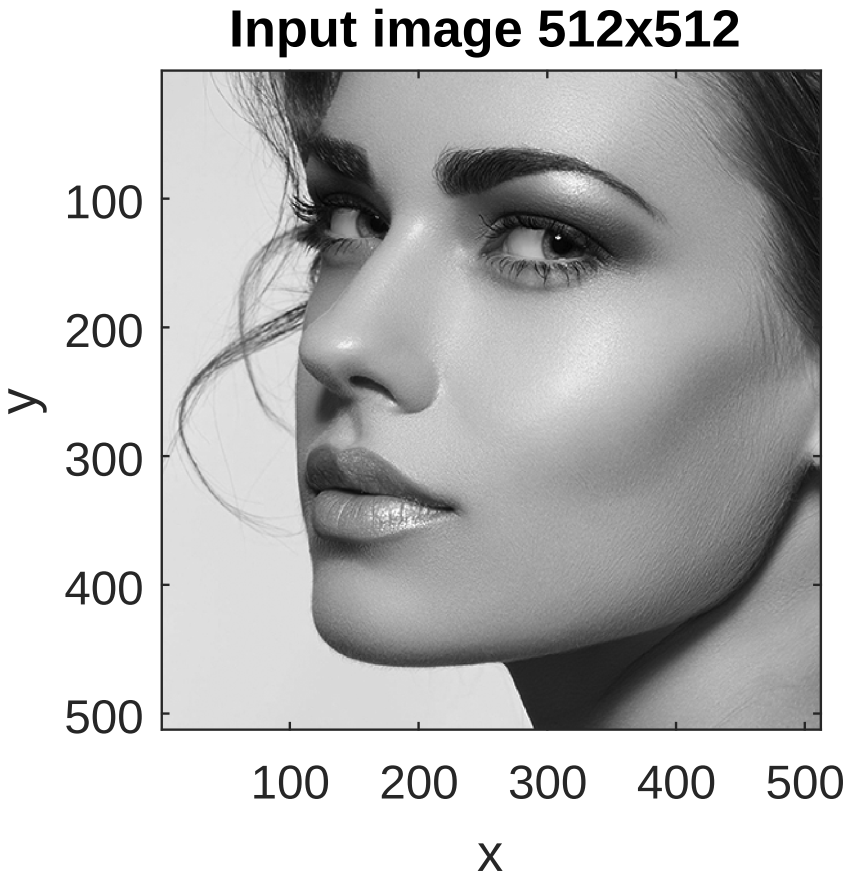


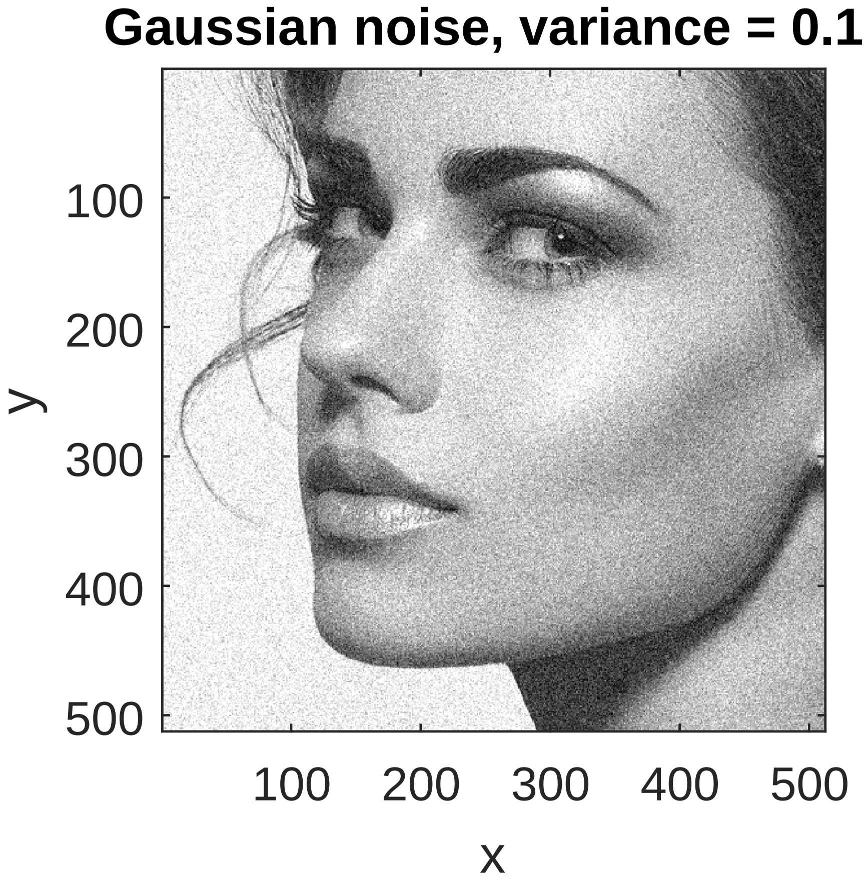
Example, Gaussian noise, variance 0.05



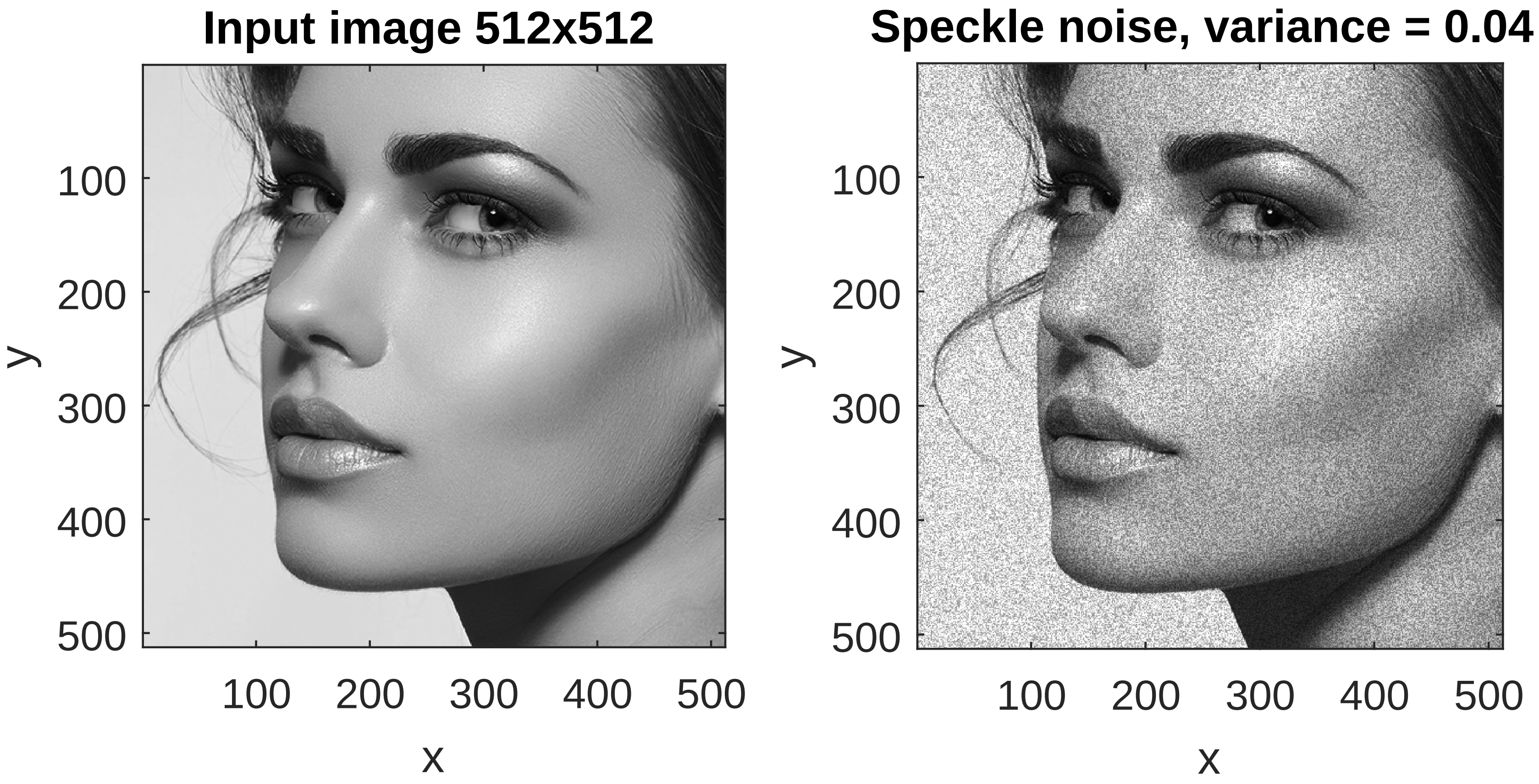


Example, Gaussian noise, variance 0.1

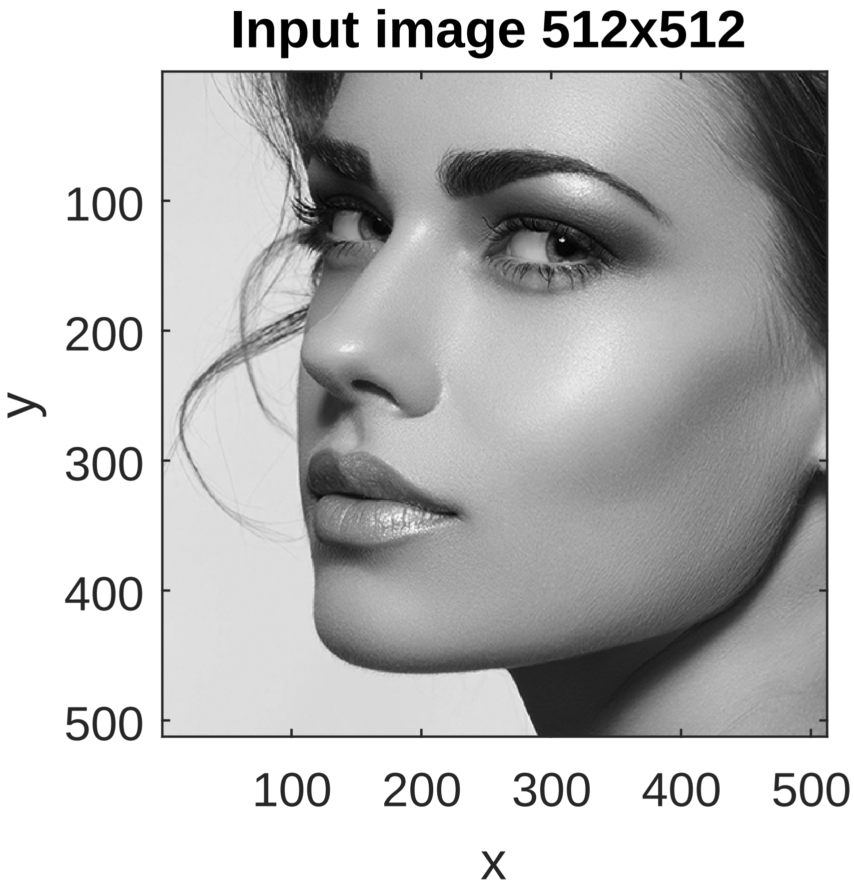


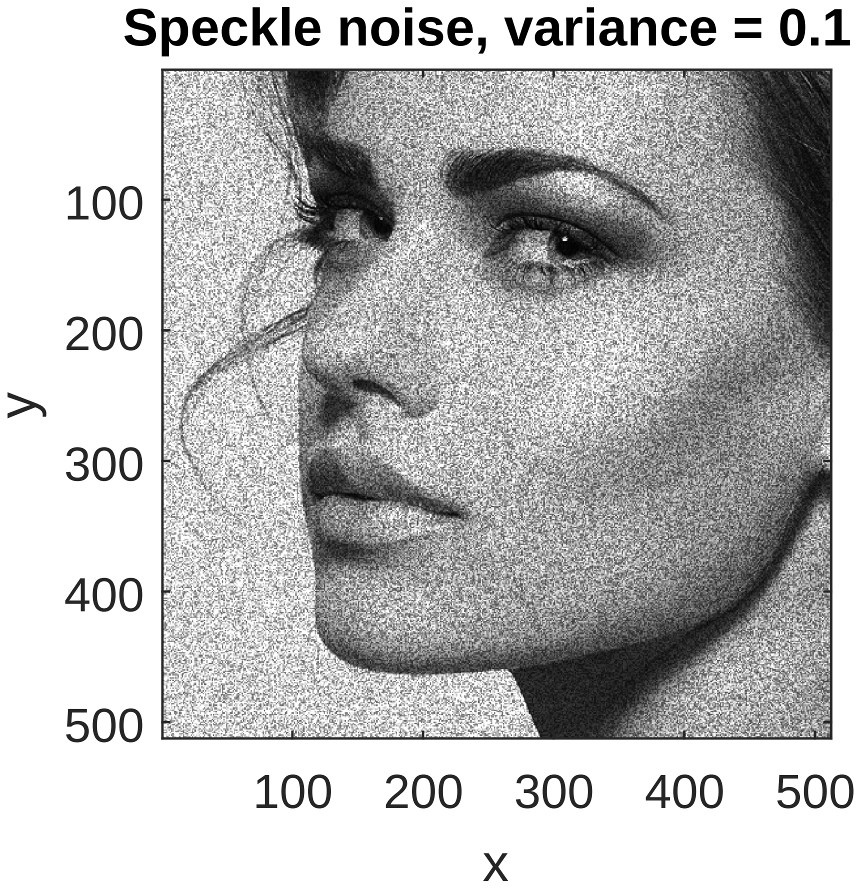


Example, speckle noise, variance 0.04



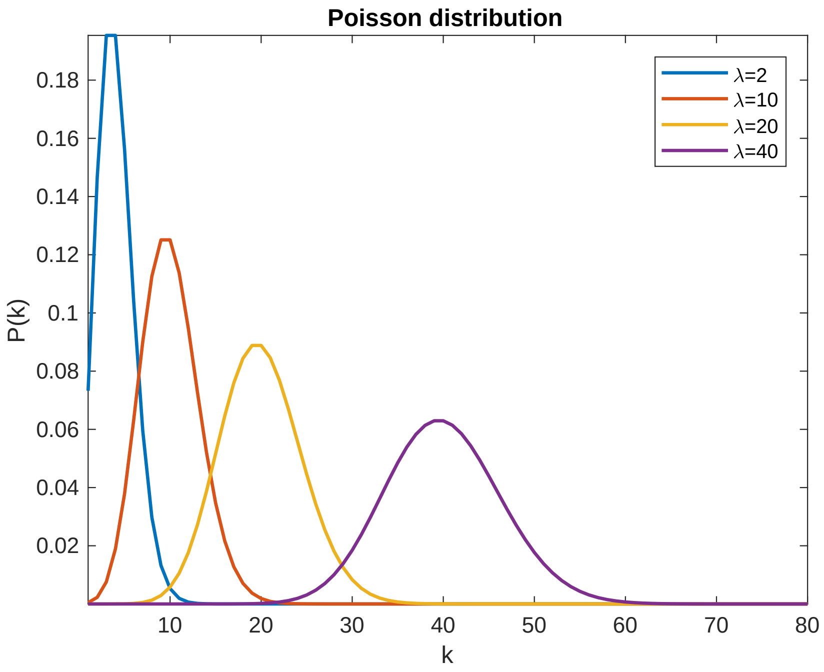
Example, speckle noise, variance 0.1





Poisson noise, continued from page 8

- Poisson noise model corresponds to a photon count, where λ is proportional to the number of photons that hits the receptor during the exposition time.
- This is useful to model medical imaging (confocal microscopy, SPECT tomography), astronomy, and digital camera noises.
- Note that λ is the mean of the distribution, but is also its variance, so that the noise intensity perturbating the image pixel $f(x,y)$ is proportional to $f_0(x,y)$.



$$P(f(x,y)=k)=\frac{\lambda^k}{k!}e^{-\lambda}$$

Statistical principle of the noise filtration

Let us consider almost the simplest image statistical model.

Assume that image function value $f(x, y)$ in each image pixel is contaminated by additive noise, i.e. $f(x, y) = g(x, y) + \nu(x, y)$, where $g(x, y)$ is the uncorrupted image and $\nu(x, y)$ is noise with the properties:

- $\nu(x, y)$ is statistically independent of the image function $g(x, y)$,
- has a zero mean μ and has a standard deviation σ.

Let observe i realizations of the image f_i, $i = 1, \ldots n$. The estimate of the correct value is

$$\frac{g_1 + \ldots + g_n}{n} + \frac{\nu_1 + \ldots + \nu_n}{n} \,.$$

The estimate outcome is a random variable with $\mu' = 0$ and $\sigma' = \sigma/\sqrt{n}$.

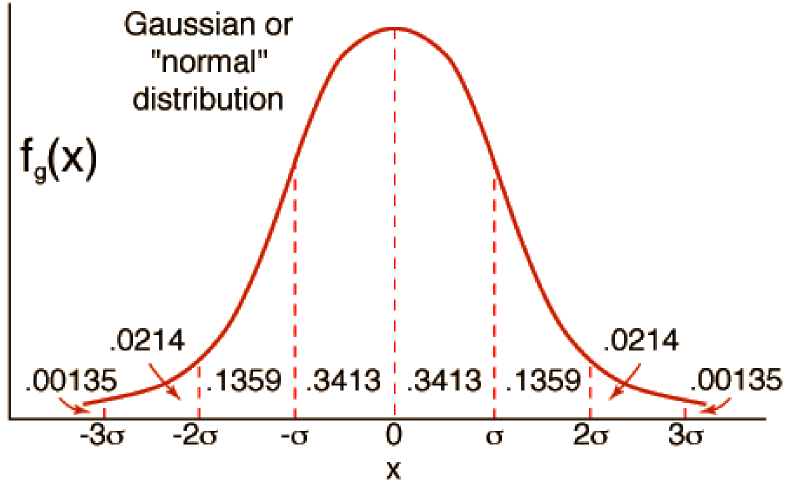
The thought above is anchored in the probability theory in its powerful Central limit theorem.

Central limit theorem (1)

- The Central limit theorem describes the probability characteristics of the ‘population of the means’, which has been created from the means of an infinite number of random population samples of size N, all of them drawn from a given ‘parent population’.
- The Central limit theorem predicts characteristics regardless of the distribution of the parent population.
- The mean of the population of means (i.e., the means of many times randomly drawn samples of size N from the parent population) is always equal to the mean of the parent population.
- The standard deviation of the population of means is always equal to the standard deviation of the parent population divided by the square root of the sample size N.
- The distribution of sample means will increasingly approximate a normal (Gaussian) distribution as the size N of samples increases.

Central limit theorem (2)

- A consequence of the Central limit theorem is that if we average measurements of a particular quantity, the distribution of our average tends toward a normal (Gaussian) one.
- In addition, if a measured variable is actually a combination of several uncorrelated variables, all of them 'contaminated' with a random error of any distribution, our measurements tend to be contaminated with a random error that is normally distributed as the number of these variables increases.
- Thus, the Central limit theorem explains the ubiquity of the bell-shaped 'Normal distribution' in the measurements domain.



Central limit theorem (3), the application view

- It is important for applications that there is no need to generate a big amount of population samples. It suffices to obtain one big enough population sample. The Central limit theorem teaches us what is the distribution of population means without the need to generate these population samples.
- What can be considered a big enough population sample? It is application dependent. Trespassing the lower bound of 30-50 random observation is not allowed by statisticians. Recall samples with about 1000 observations serving to estimate outcomes of elections.
- The confidence interval in statistics indicates the reliability of the estimate. It gives the degree of uncertainty of a population parameter. We have talked about sample mean only so far.

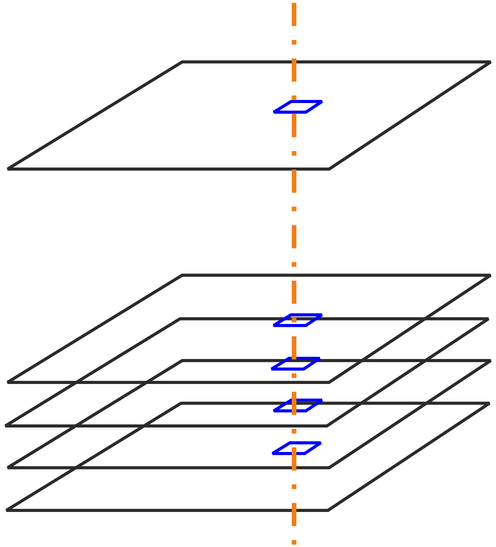
See a statistics textbook for details.

Smoothing from several images without blurring

Assumptions: n images of the same unchanged scene, in which it can be assumed that noise is independent to the image.

The correct intensity value: $f(i,j)$ is estimated from a random population given by all pixels at the same position in all input images $g_k(i,j)$, e.g., by ordinary averaging,

$$f(i,j) = \frac{1}{n} \sum_{k=1}^{n} g_k(i,j) \, .$$



n stacked images

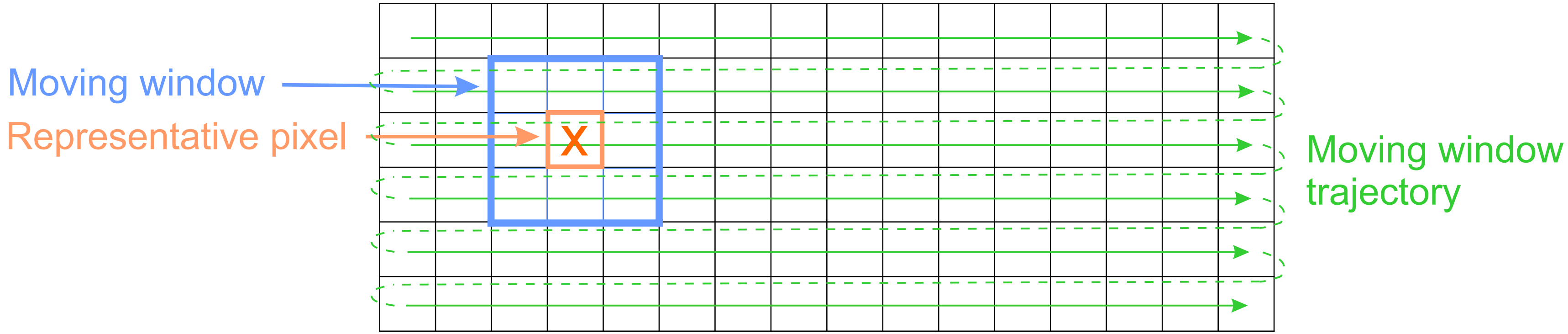
Example: Suppression of thermal noise for cameras utilized for precise measurements. The correct value is usually estimated from 50 images at least.

The trouble: need to filter noise from a single image

- No other choice, but to resort to the data redundancy in the image.
- Neighboring pixels have the same or similar intensity value.
- The intensity value can be corrected based on the analysis of intensities in the neighborhood. A single typical sample or the combination of several intensity values in the neighborhood is taken.
- The trouble with blurring on the steep intensity transitions occurs..

General local filtration

- The correct (new output) intensity value is estimated from the image function values in a small neighborhood $\mathcal{O}$ of the current (representative) pixel. This neighborhood is often a small rectangle, called also a filtering window.
- Imagine that the image is systematically traversed line by line, e.g. from top left.



- The result of the image function analysis is written to the output image at the position of the representative pixel in the input image. The method is (sometimes) called Moving window filtration.
- In general, the filter properties can vary with the position of the representative point.

Operators independent to the shift, called also the space invariant filters

- It is a special case of the local filtration.
- The properties of the filter remain the same in all positions of the representative point in the image.
- These filters have a counterpart in the frequency domain filtration, e.g. in the Fourier spectrum.

Local linear preprocessing

- The current value is calculated as a linear combination (convolution) of the image function values in the local neighborhood.
- Reminder: the operator linearity, i.e. two properties: additivity and homogeneity.
- The linearity assumption is violated for real images
  - There is a problem of limit pixel values (intensities), as intensities lie i.e. typically in the interval $\langle 0, 255 \rangle$.

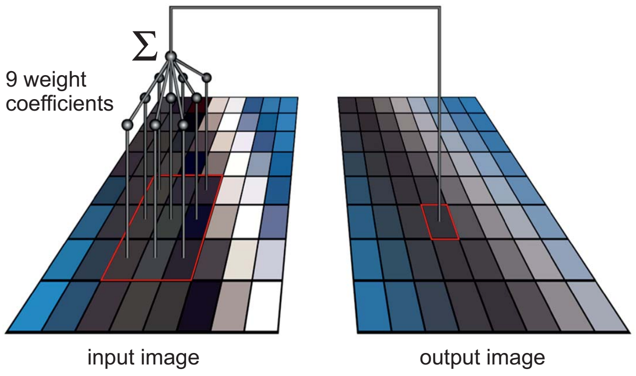
    The pixel value cannot be multiplied by an arbitrary non-zero number because the result would lie outside the limits. Similarly for addition of two pixel values.
  - Troubles at the image borders.

A discrete 2D convolution

- The contribution of individual pixels in the neigborhood $\mathcal{O}$ is weighted by coefficients h in the linear combination

$$g(x,y) = \sum_{(m,n)} \sum_{\in \mathcal{O}} h(x-m,\, y-n)\; f(m,n)\,.$$

- The convolution kernel h, also convolution mask.
- A rectangular neighborhood $\mathcal{O}$ with odd number of rows and columns is often used in order to have a representative point in the middle of the mask.



Convolution vs. cross-correlation

- Convolution $*$ *(related to the impulse response)*

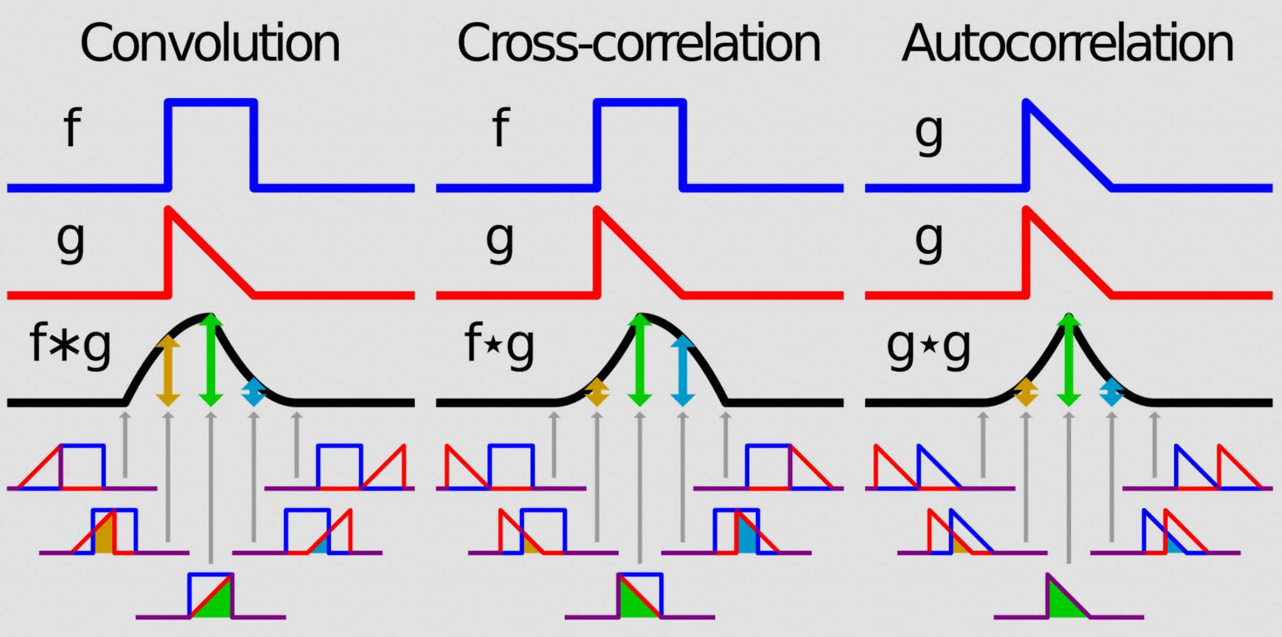
$$g(x,y) = (h * f)(x,y) = \sum_{(m,n)} \sum_{\in \mathcal{O}} h(x-m,\, y-n)\; f(m,n)\,.$$

- Cross-correlation $\star$ *(a measure of similarity between two signals/images)*

$$g(x,y) = (h \star f)(x,y) = \sum_{(m,n)} \sum_{\in \mathcal{O}} h(x+m,\, y+n)\; f(m,n)\,.$$

- Convolution by cross-correlation:
  - Flip filter h in both dimension (bottom to top, left to right).
  - Perform cross-correlation.
- The result of convolution and correlation is the same for symmetric filters.

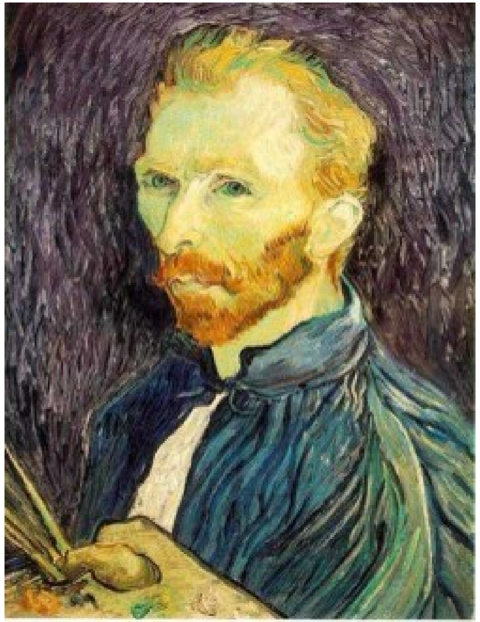
Convolution vs. cross-correlation, 1D illustration



Courtesy: commons.wikimedia.org

Linear filtration in spatial domain; the example

$=$



$*$

$$h(x, y)$$

filtered image g

original image f

convolutional kernel h

also filter function, here

Gaussian $h(x, y) = e^{-\frac{x^2+y^2}{\sigma^2}}$

Ordinary averaging

Averaging in a 3×3 neighborhood

$$h = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} .$$

Modifications, which stress importance of pixels close to the center of the mask

$$h = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} , \quad h = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} .$$

Ordinary averaging, no noise example

Original 512×512

Averaging 3×3

Averaging 7×7

Averaging 11×11

Averaging 25×25

Ordinary averaging, Gaussian noise example

Additive Gaussian noise, $\sigma = 0.05$

Averaging 3×3

Averaging 7×7

Original 512×512

Averaging 11×11

Averaging 25×25

Ordinary averaging, salt & pepper noise example
5 % pixels corrupted

Add. salt & pepper noise, $\sigma = 0.05$

Averaging 3×3

Averaging 7×7

Original 512×512

Averaging 11×11

Averaging 25×25

Separable filters

- Filter is separable if $h(x, y) = h_1(x) \cdot h_2(y) = \delta * h_1(x, \cdot) * h_2(\cdot, y)$, where δ is a Dirac function.
- Separable filters can be implemented via successive 1D convolutions (associativity of convolution). Let M, N be size of the image, m, n be size of the rectangular convolution mask h. Separability reduces the computational complexity of the convolution from $\mathcal{O}(MNmn)$ to $\mathcal{O}(MN(m + n))$.
- Example: a binomical 2D filter of the size 5×5. The filter element is constituted as the edition of two preceding elements in Pascal triangle.

$$\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} = \begin{bmatrix} \\ \\ h_1 \\ \\ \\ \end{bmatrix} \begin{bmatrix} \quad & h_2 & \quad \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}.$$

Separability $\Rightarrow$ savings in calculations

The size of the convolution mask is $2N + 1$.

$$
\begin{aligned}
g(x,y) &= \sum_{m=-N}^{N} \sum_{n=-N}^{N} h(x-m,\, y-n)\, f(m,n) \\
&= \sum_{m=-N}^{N} \sum_{n=-N}^{N} h(m,n)\, f(x+m, y+n) \\
&= \sum_{m=-N}^{N} h_1(m) \sum_{n=-N}^{N} h_2(n)\, f(x+m, y+n)
\end{aligned}
$$

Separability $\Rightarrow$ savings in calculations 2

- Our filter of the size 5×5 needs 25 multiplications and 24 additions in each pixel.
- If a separable filter is used, only 10 multiplications and 8 additions.

- The saving would be more dramatic in the case of the convolution-based filtration in the 3D image, e.g. from a tomograph. For the convolution mask of the size $9 \times 9 \times 9$, there is a need for 729 multiplications and 728 additions.
- 27 multiplications and 24 additions suffice per voxel.

Separability check and related decomposition

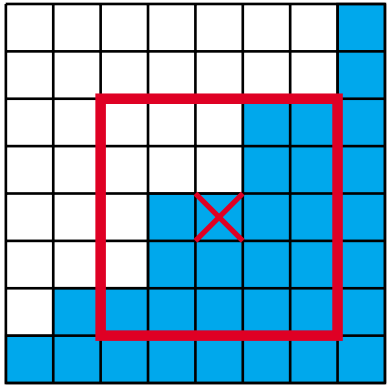
Each filter with the rank 1 is separable.

The singular decomposition (SVD).

```
[u,s,v] = svd(A);
s = diag(s);
tol = length(A) * max(s) * eps;
rank = sum(s > tol);
if (rank == 1)
  hcol = u(:,1) * sqrt(s(1));
  hrow = conj(v(:,1)) * sqrt(s(1));
  y = conv2(hcol, hrow, x, shape);
else
  %Nonseparable stencil
end
```

Nonlinear smoothing

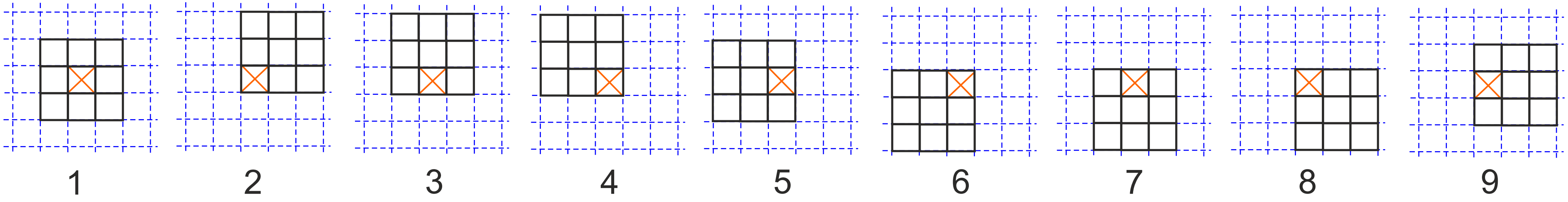
- The aim: to reduce the blur of edges while smoothing.
- Two possible principles to estimate the correct value of image function while less damaging edges:
  - First principle: find such a subset of the current pixel neighborhood, in which the intensity does not change much.



  - Second principle: The mean value is a bad estimate if outliers are present. Use the robust statistics instead of it.

Rotating mask method

- The homogeneous part of the 5×5 is sought by a rotating mask of the size 3×3.
- There are nine possible positions of the 3×3. One position is in the middle and eight additional positions are illustrated in the image. The orange cross denotes the representative pixel of the filtration mask.



- The mask with the minimal dispersion of intensity is selected for estimating the correct value in the current mask position given by the representative point.

Median filtration

- Median $=$ one of quantiles of the population, 2-quantile.

  Quantiles are points taken at regular intervals from the cumulative distribution function of a random variable (We know it from the histogram equalization). $q-$quantile divides ordered data into q equal-sized data subsets.

- Let x be a random variable. Median M is the value x for which the probability of the phenomenon $x < M$ equals to one half.

- Calculating the median for a discrete image function is simple. It suffices to order intensity values in a local neighborhood. The median is given by the element which is in the middle of the neighborhood.

- The series with the odd number of elements are often used in order to determine uniquely the position in the middle of the series. 3×3, 5×5, atd.

- The calculation can be sped up because it can be shown that only partially ordered series suffices for determining the median.

Median, the example

100	98	102
99	105	101
95	100	255

- The orange box denotes the representative pixel of the filtration mask (neighborhood).
- Mean = 117,2
- Median : 95 98 99 100 100 101 102 105 255
- The median calculation is robust because it copes with up to 50 % outliers.

Median filtration, Gaussian noise example


Additive Gaussian noise, $\sigma = 0.05$


Averaging 3×3


Averaging 7×7


Original 512×512


Median filtration 3×3


Median filtration 7×7

Median filtration, salt & pepper noise example 5 % pixels corrupted

Add. salt & pepper noise, $\sigma = 0.05$

Averaging 3×3

Averaging 7×7

Original 512×512

Median filtration 3×3

Median filtration 7×7