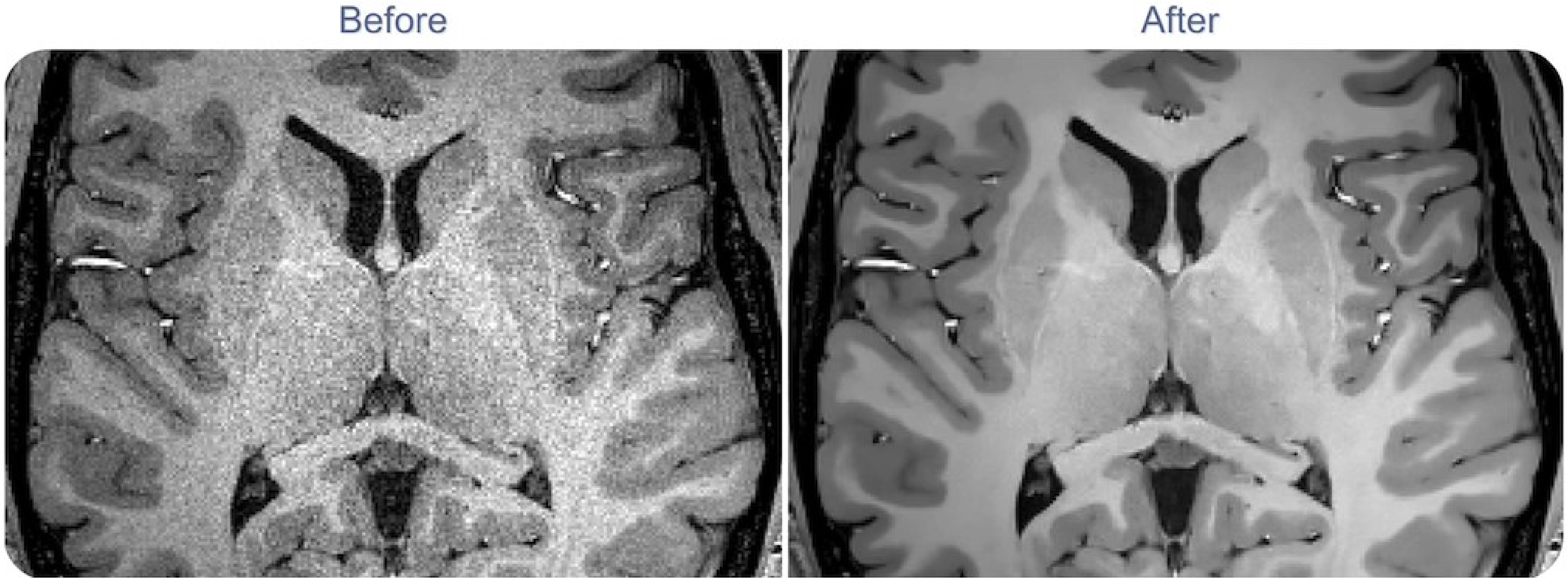
CS463/516

Image enhancement

Image enhancement

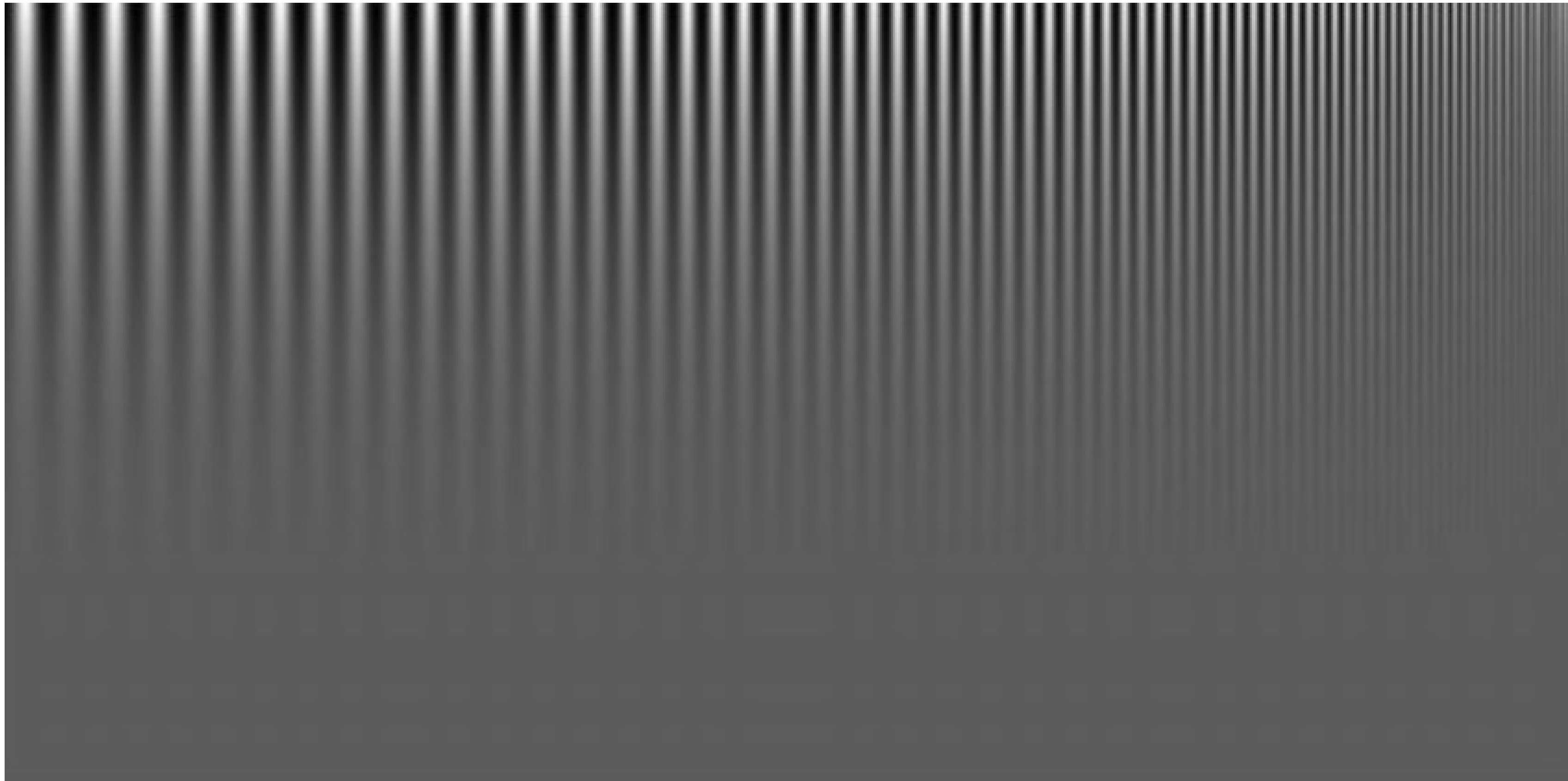
- Two reasons for image enhancement
  - 1) increases perceptibility of objects in image for human observer
  - 2) preprocessing step for subsequent automated image analysis (segmentation or other)
- Typically, enhancement reduces artifacts/noise, and emphasizes difference between objects of interest
- Need quantitative measures of image quality to know if enhancement succeeded

Example: non-local means algorithm employed to remove noise from a T1-weighted MRI image.



Spatial resolution and contrast

- Spatial resolution determines the smallest structure that can be represented in a digital image
  - No detail with frequency < 2x the sampling distance can be represented without aliasing (Nyquist theorem)
- *Perceived* resolution may be measured experimentally by treating human visual system as black box
- Example: test pattern for determining perceived resolution in line pairs per mm



- Number of line pairs per mm increases from left to right
- Contrast decreases from top to bottom
- Perceived resolution depends on the contrast in the image, and the relationship is nonlinear

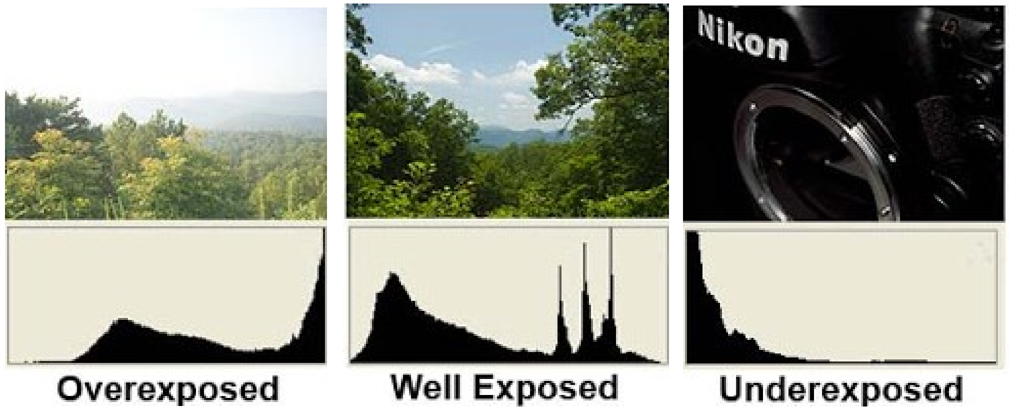
Definition of contrast

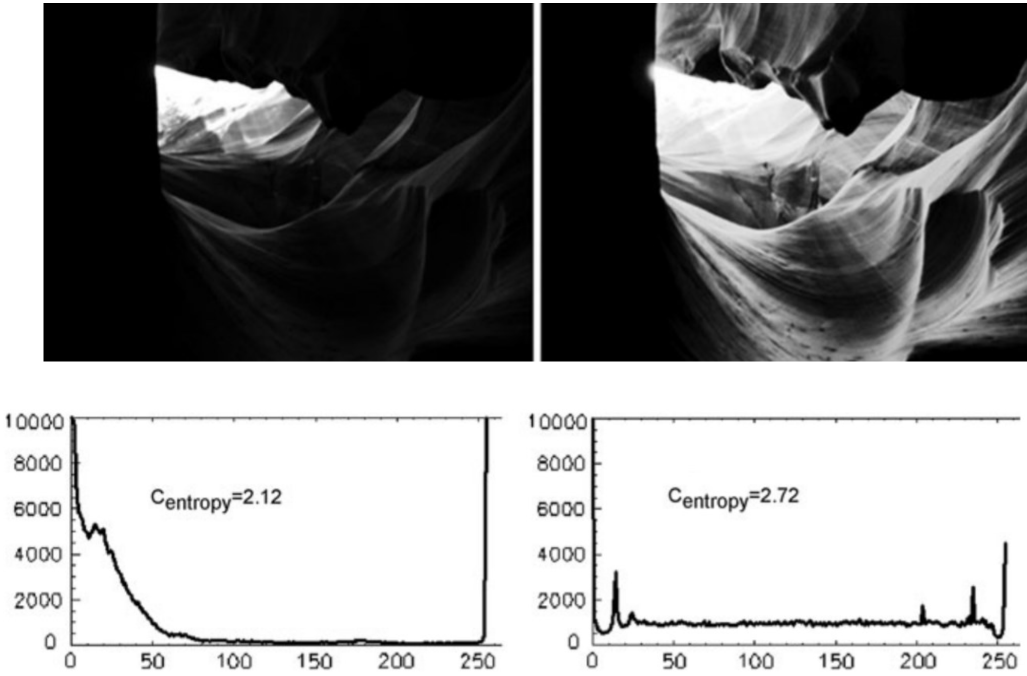
- To determine contrast, need to know what is an object and what is background
- This is typically unknown prior to analysis, so object-independent contrasts are defined:
- Michelson contrast $C_{Michelson}$ measures utilization of the luminance range
  - Smallest luminance is $l_{min} = 0$ (no luminance), upper boundary is arbitrarily high
  - If smallest displayed luminance is f_{min} and largest is f_{max}, Michelson contrast is: $C_{Michelson} = \frac{f_{max}-f_{min}}{f_{max}+f_{min}}$
    - Michelson contrast ranges from 0 to 1
  - Does not account for *distribution* of intensities in image
- Root-mean-square (rms) contrast:
  - Given image $f(x,y)$ with intensities $I(x,y)$ the expected value of i is $\bar{I} = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{J=0}^{N-1} I(i,j)$ and rms contrast is given by:
  - $C_{rms}(f) = \sqrt{\frac{1}{MN-1}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(I(i,j) - \bar{I})^2}$
  - C_{rms} takes all pixels into account, but still doesn't differentiate well between different intensity distributions
  - Example: two images at right have same C_{rms} (and same $C_{Michelson}$)

Entropy as a contrast measure

- histogram
- Entropy as a contrast measure includes histogram characteristics into measure
- Computed from normalized histogram of image intensities
- Histogram $H(l)$ of an image $l(x, y)$ gives frequency of occurrence for each intensity value
- A *normalized histogram* $H_{norm}(l)$ is computed from $H(l)$ by: $H_{norm}(l) = \frac{H(l)}{\sum_{k=I_{min}}^{I_{max}} H(k)}$
- Which gives the probability of l to appear in an image. Increased entropy = enhanced contrast

3 separate images and their histograms



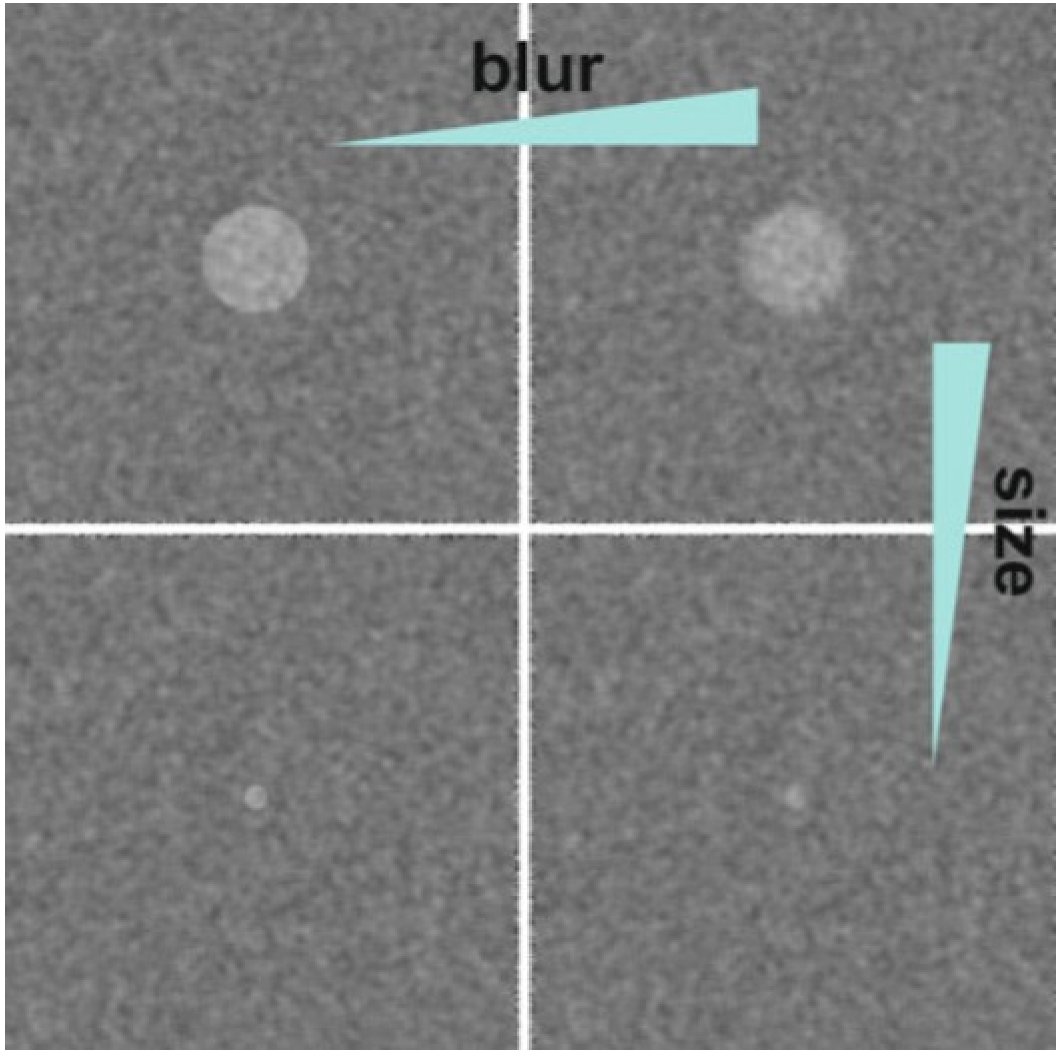


Signal-to-noise ratio

- *Noise* in image is another factor limiting perceptibility of objects in image
- Noise $n(i,j)$ in image usually described as random fluctuations of intensities with zero mean
- If noise is assumed to be normally distributed with zero mean, variance $\sigma^2(n)$ characterizes the noise level
- a) four images have same noise level, noise characteristics, and contrast. *Object-dependent* features such as size of object or sharpness of boundaries cause differences in perceptibility of depicted objects
- Object detection depends on ratio of object-background contrast to noise variance, giving the *Signal-to-noise ratio* (SNR): $SNR(f) = S(f)/\sigma(n)$
- In simplest case, $S(f)$ defined to be f_{max} (largest intensity in image) or the mean intensity
- Common measure given in dB:

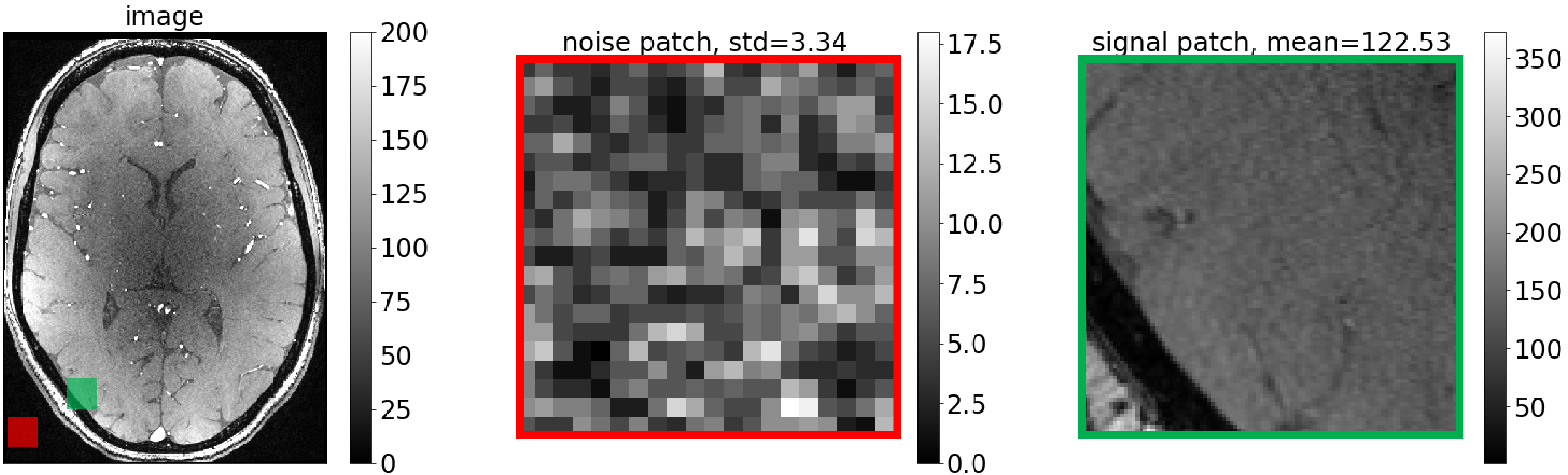
$$SNR_{dB} = 10\log_{10}\left(\frac{1}{MN}\frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(f(i,j)-\bar{f}\right)^2}{\sigma^2}\right)$$

a)



Noise and SNR in MRI

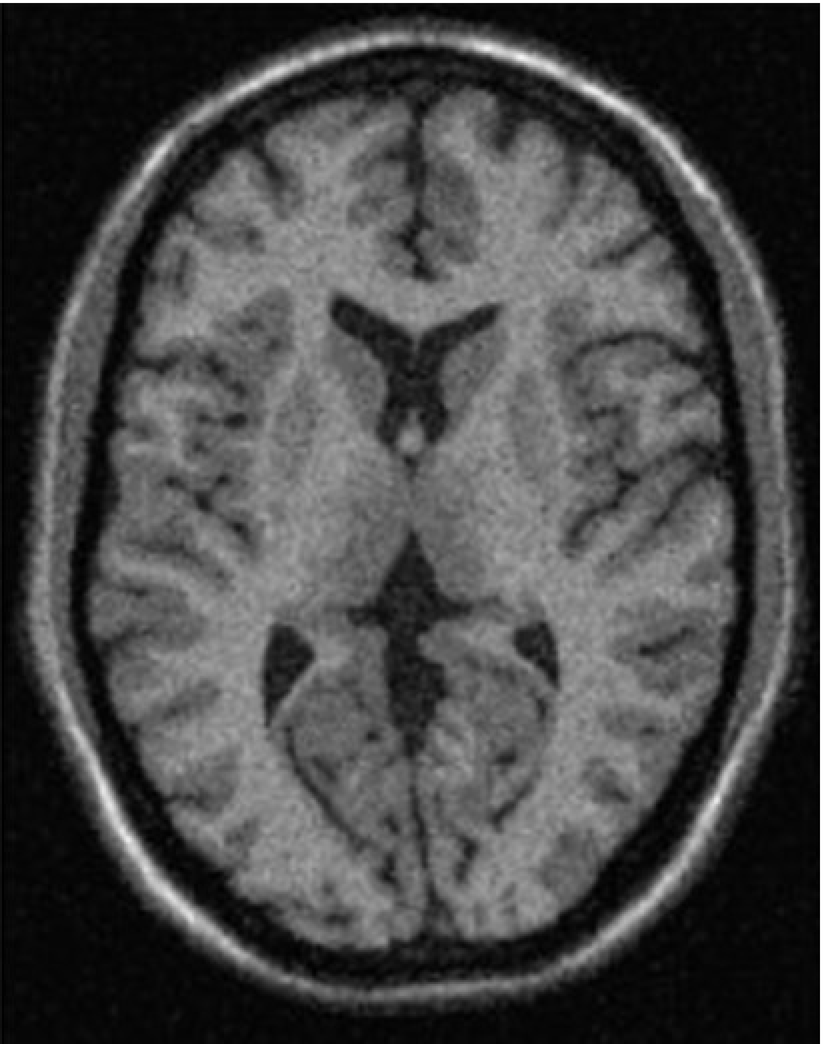
- $SNR = mean(Signal)/std(noise)$
  - Where the noise is assumed to be gaussian



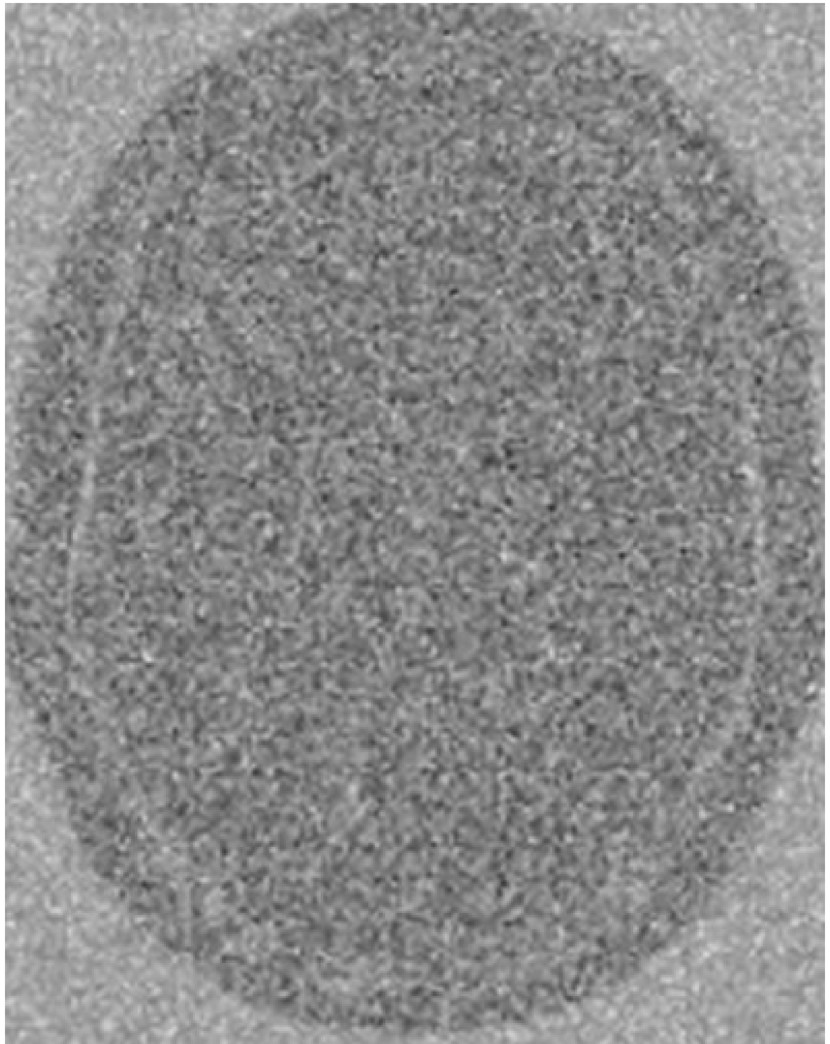
$$\frac{122.53}{3.34} = 36.68 = \text{SNR}$$

Noise reduction

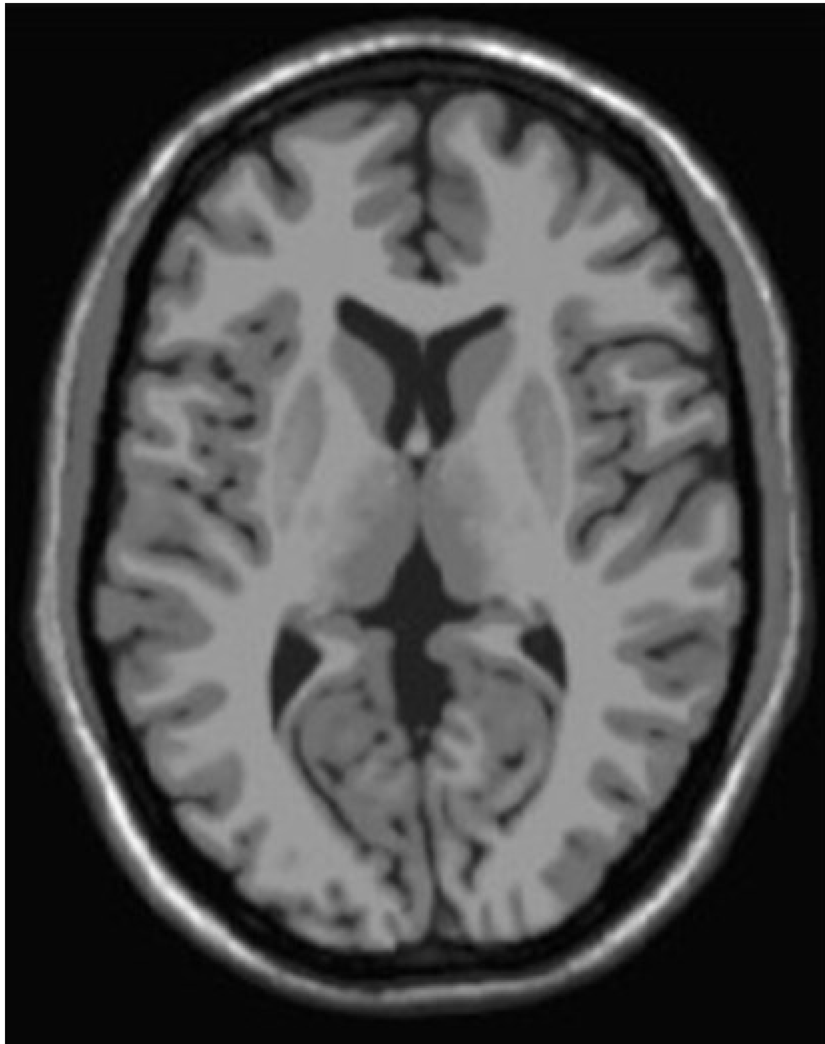
Noisy image g



Noise n



Noiseless image: $f = g - n$



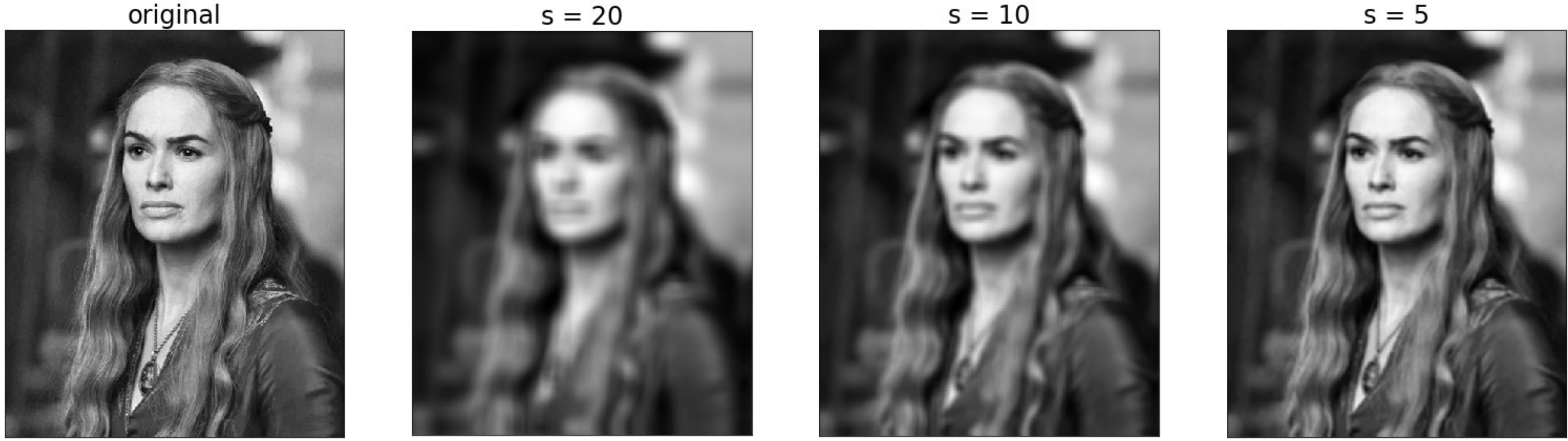
Noise reduction

- Noise usually modeled as stationary and additive with zero mean
- Noisy image g related to noise-free image f through $g = f + n$, where n is zero-mean noise
- Noise removal through linear filtering consists of estimating the expected value $E(g)$. Since $E(n) = 0$, we have: $E(g) = E(f) + E(n) = E(f) = f$ because deterministic function f has $E(f) = f$.
- Noise reduction schemes try to reconstruct $E(g)$ from various assumptions about f:
  - **Linear filtering** assumes f to be locally constant. E then estimated by averaging over local neighborhood
  - **Median filtering** assumes noise is normally distributed, f is locally constant, except for edges, the signal at edges is higher than noise, and edges are locally straight
  - **Diffusion filtering** and their approximations assume that f is locally constant, except at edges, and properties of edges and noise can be differentiated by amplitude or frequency
  - **Bayesian image restoration** requires f to be locally smooth, except for edges. Further requires that in some local neighborhood, edge pixels are not the majority in that neighborhood
- Most of these assumptions are not true everywhere in the image, so filtering results in various filter-specific artifacts

Noise reduction by linear filtering

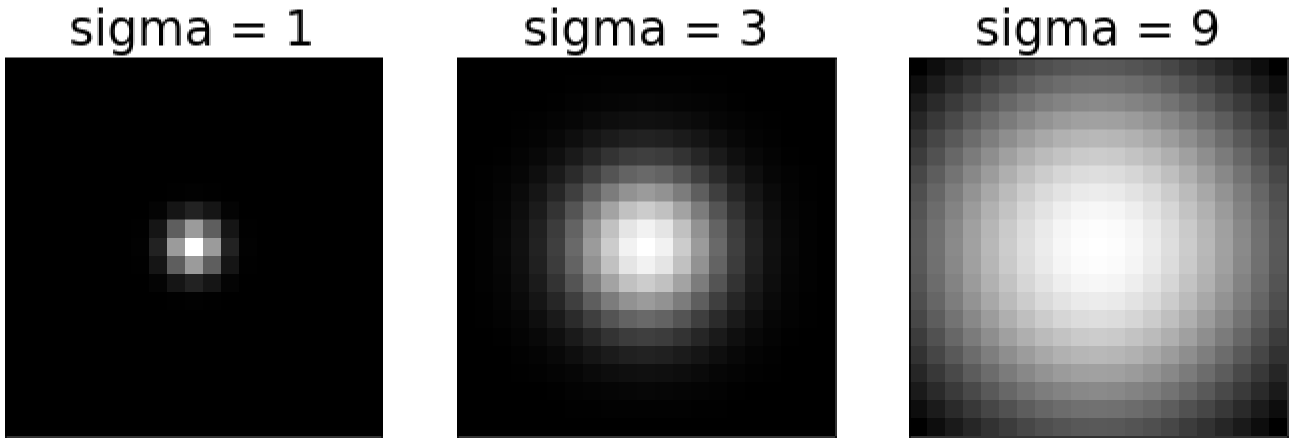
- If f is constant in some neighborhood around (i,j) then $E(i,j)$ can be estimated by averaging over this neighborhood
- This operation carried out in spatial domain as convolution with a *mean filter* of size s: $f(i,j) \approx [g * c_{mean,s}](i,j)$
  - Where $*$ stands for the convolution operation
  - The convolution kernel $c_{mean,s}$ is a square matrix of size $s \times s$, with s being odd
- The estimate of $E(g(i,j))$ improves with size of s but the likelihood that f is constant in the neighborhood for all locations (i,j) decreases, leading to increased blurring at edges

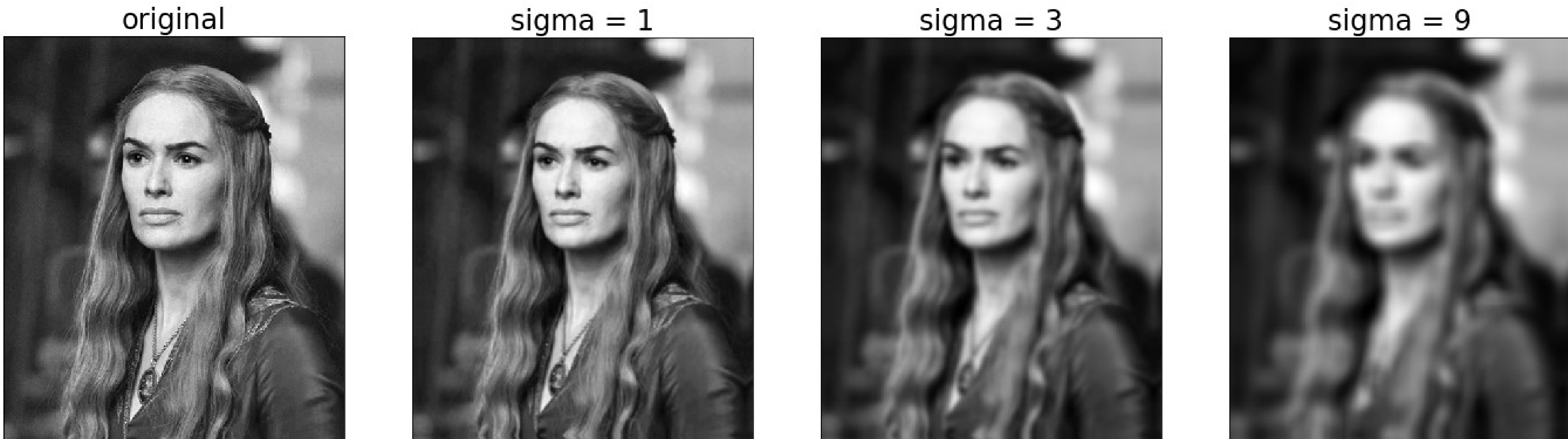
$$c_{\text{mean},s} = \frac{1}{s^2} \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & & \dots & 1 \\ \dots & & & \dots \\ & & 1 & 1 \end{pmatrix}$$



Noise reduction by linear filtering (Gaussian)

- $G(x, y) = \frac{1}{2\pi\sigma^2} \mathrm{e}^{-\frac{\mathrm{x}^2+\mathrm{y}^2}{2\sigma^2}}$
- Filtering may also be carried out by multiplication in the frequency domain
  - Assignment 1, part 2b,c
- The gaussian for filtering in frequency space corresponds to a Gaussian in the spatial domain with inverted standard deviation

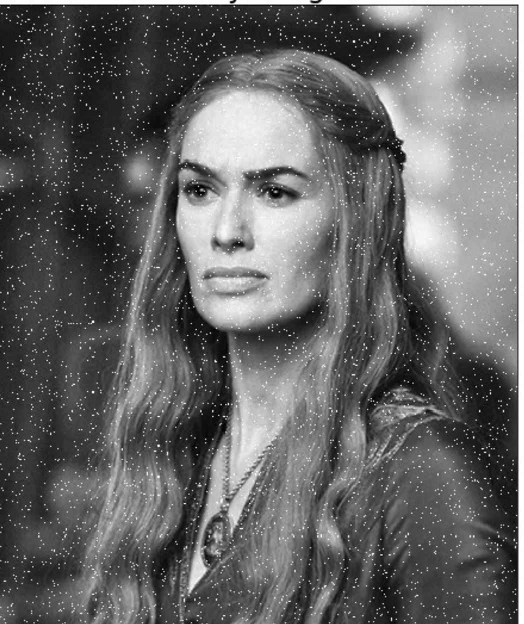




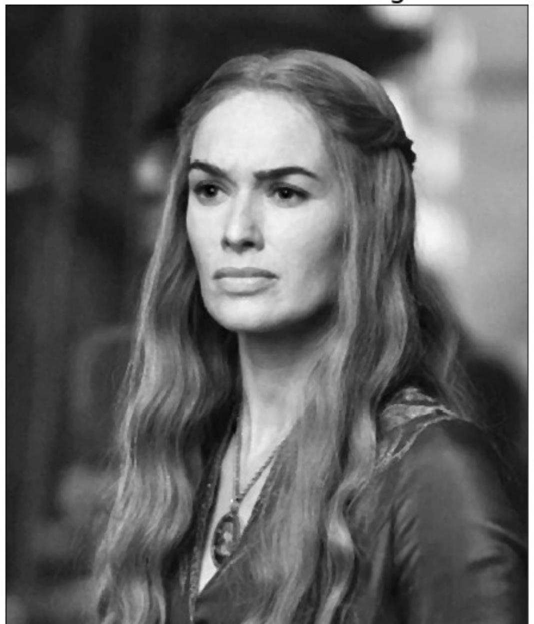
Median filtering

- Linear filtering produces poor results if SNR is low or low frequency noise is high
- Successful noise reduction in such cases requires an edge model as part of the smoothing process, such methods are called *edge-preserving smoothing*
- **Median filtering** is a simple way to achieve edge-preserving smoothing
- Median filter sorts the values in a neighborhood, and replaces pixel in middle of neighborhood with the median
- Since the median filter does not average over the neighborhood, it is particularly suitable to remove outliers
- Under certain conditions, the median filter also preserves edges

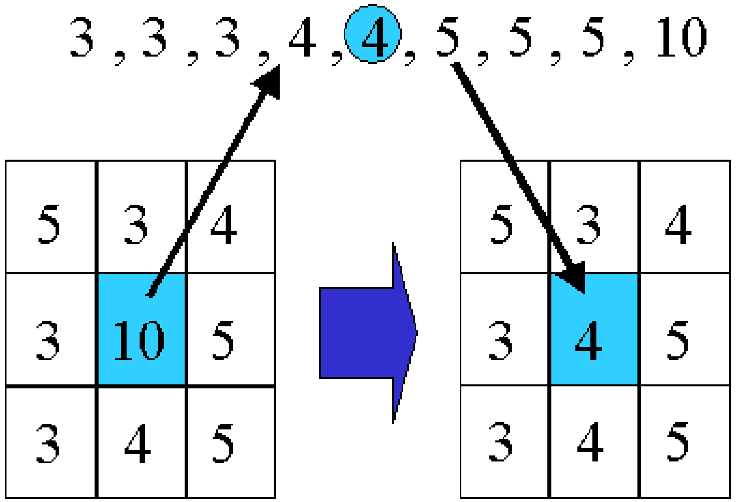
noisy image



median filtered image

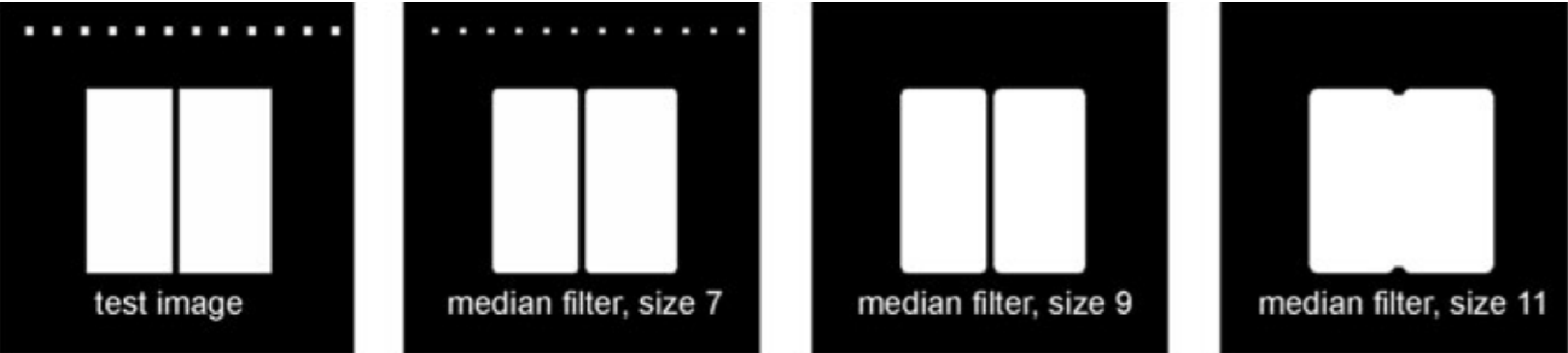


gaussian filtered image (sigma=3)



Median filter implicit edge model

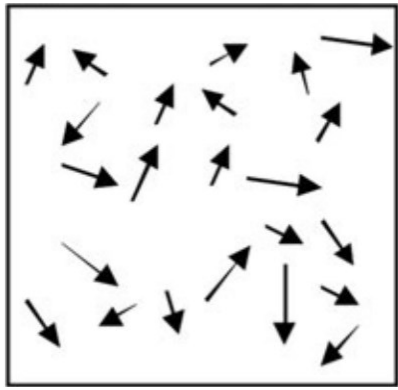
- Median filter preserves edge under following conditions:
  - 1) the edge is straight within the neighborhood of the filter
  - 2) signal difference of two regions incident to edge exceeds noise amplitude
  - 3) signal is locally constant within each of the two regions incident to edge
- Example: median filtering on simple test image.
  - Corners are rounded off by filters of any size, since no neighborhood size exists in which boundary is locally straight at corners
  - Object details are removed if the filter size is larger than the detail
- Some variants exist that adapt the filter size to local image statistics



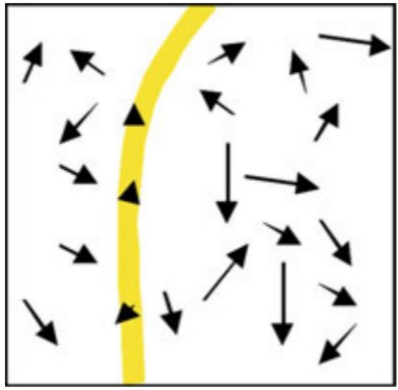
Diffusion filtering

- Median filter has two disadvantages:
  - 1) doesn’t remove noises at edges, even if edge follows implicit edge model
  - 2) may alter edges in a random fashion that does not follow the edge model
- Diffusion filtering uses diffusion of liquid or gaseous material as a model for noise reduction
- Image intensity is modeled as material density
- Noise is taken as density variation and diffusion is carried out iteratively
- a) *homogenous diffusion* levels any density inhomogeneity, resulting in a noise-free image without edges
- Diffusion across edges should be inhibited for edge enhancement.
  - Boundaries are unknown, so edge response from an edge-enhancement algorithm is used to find potential boundary locations
- b) *inhomogeneous diffusion* treats such boundary locations as a semi-permeable material (image intensities will be blurred less at edges)
- c) *anisotropic diffusion* – allowing diffusion parallel to an edge, while inhibiting diffusion across the edge
- Gradient direction is used as discriminative feature between noise and edges
  - gradients between adjacent edge pixels have similar direction (not true for adjacent noise)

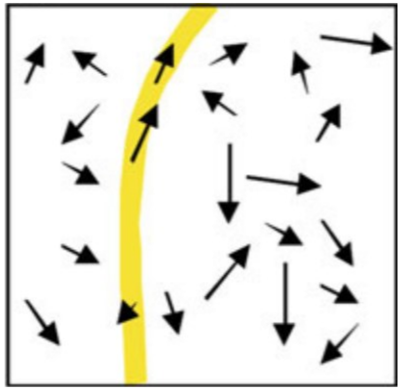
a)



b)



c)



Diffusion filtering

- Comparison of different types of diffusion filtering
- a) homogenous diffusion b) inhomogenous diffusion c) anisotropic inhomogenous diffusion

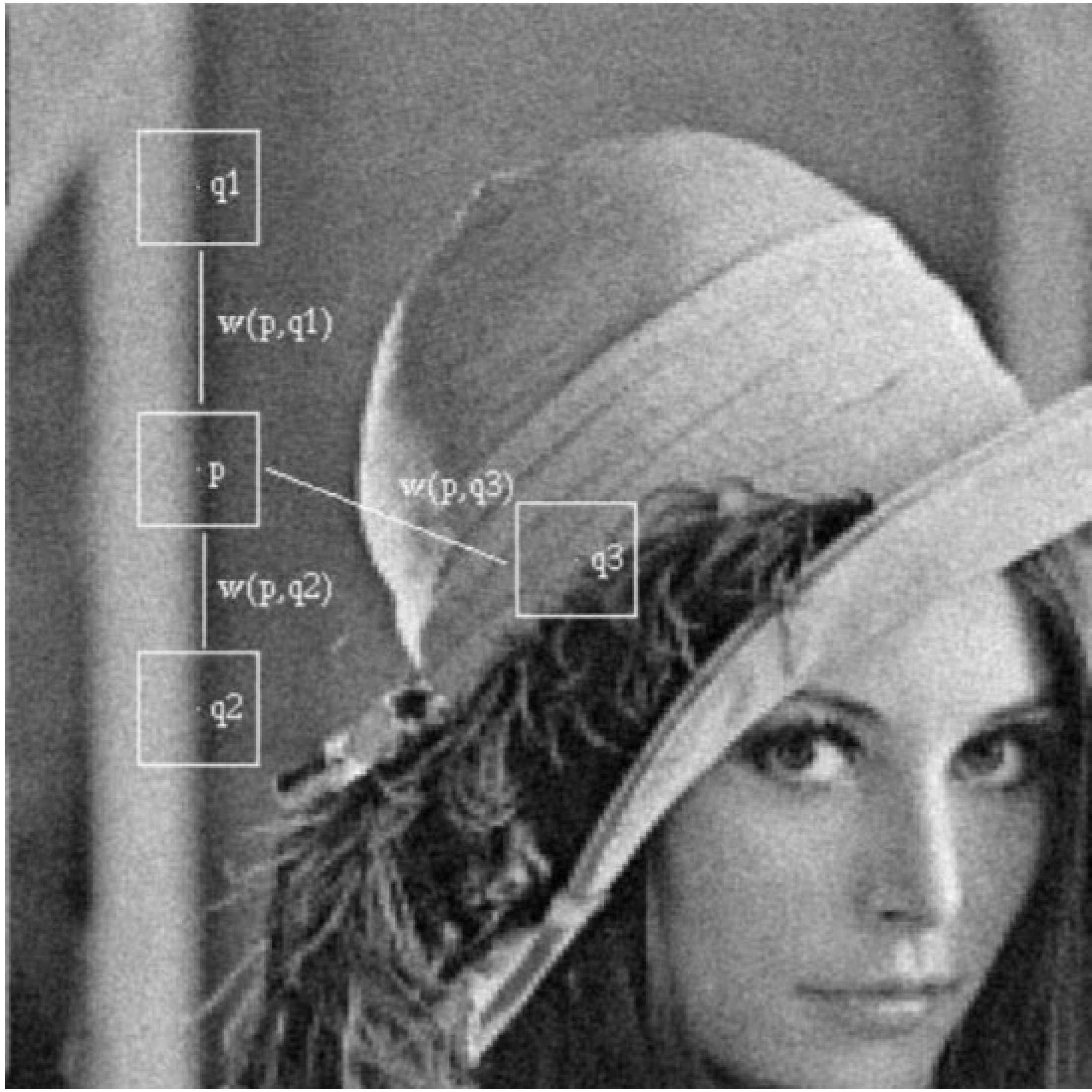
Non-local means

- Reminder: goal of image denoising is to recover original image from noisy measurement: $v(i) = u(i) + n(i)$
  - Where $v(i)$ is observed value, $u(i)$ is 'true' value, and $n(i)$ is noise at pixel i
- The basic approach common to all methods so far is *local averaging* using a neighborhood – e.g. gaussian filter defined over a 9x9 neighborhood
- **Non-local means:** given a noisy image $v = \{v(i) \mid i \in I\}$, the estimated value $NL[v](i)$, for a pixel i, is computed as a weighted average of *all pixels in the image:*
- $NL[v](i) = \sum_{j \in I} w(i,j)v(j)$ **,** where the family of weights $\{w(i,j)\}_j$ depends on the similarity between pixels i and j, and satisfy usual conditions $0 \leq w(i,j) \leq 1$ and $\sum_j w(i,j) = 1$

Non-local means

- Similarity between two pixels i and j depends on the similarity of the intensity gray level vectors $v(\mathcal{N}_i)$ and $v(\mathcal{N}_j)$, where $\mathcal{N}_k$ denotes a square neighborhood of fixed size, centered at pixel k.
- Similarity is measured as a decreasing function of the weighted Euclidean distance, $\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|_{2,a}^2$ where $a > 0$ is the standard deviation of the Gaussian kernel
- Pixels with a similar gray level neighborhood to $v(\mathcal{N}_i)$ have larger weights in the average (a)

a)



Non-local means

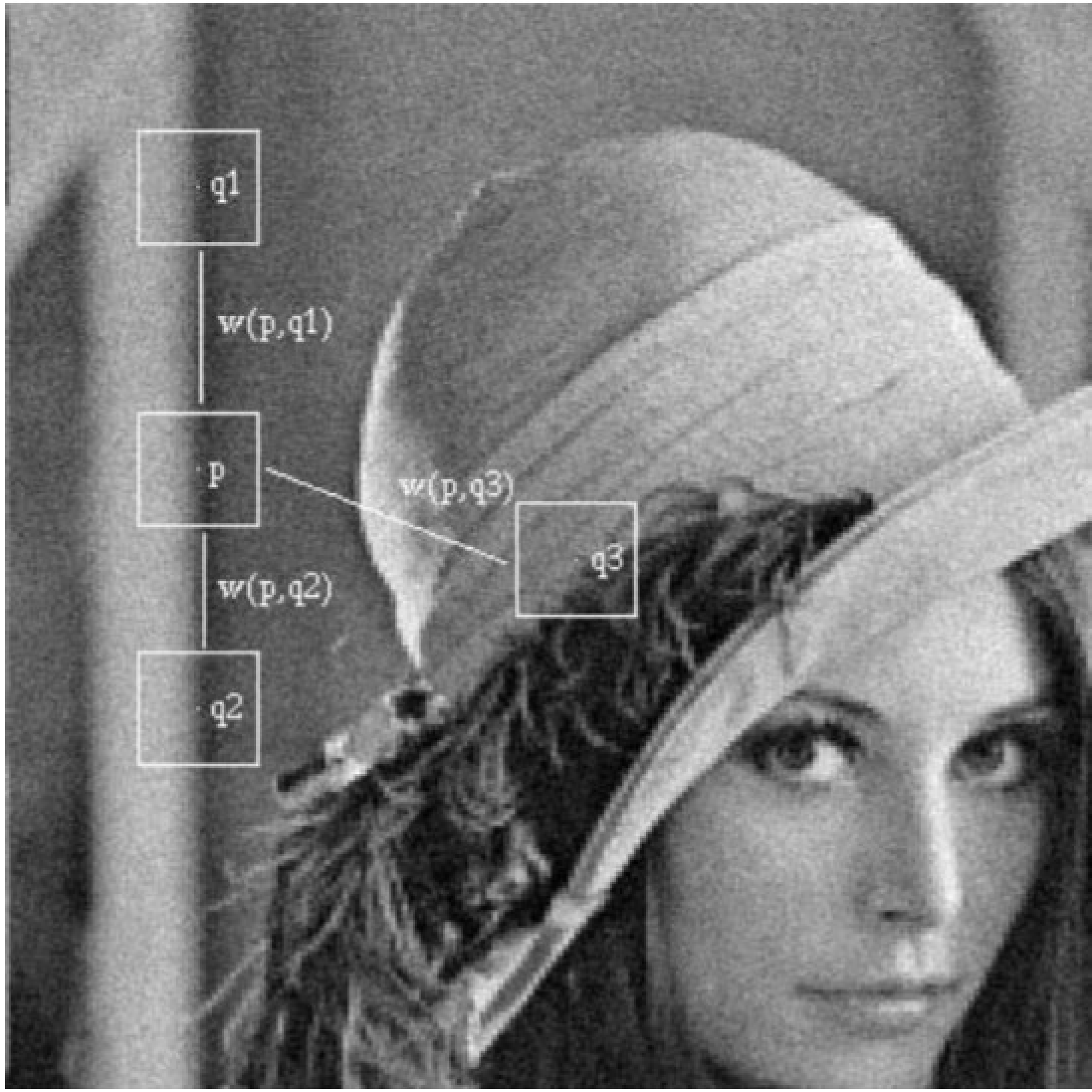
- Weights are defined as:

$$w(i,j) = \frac{1}{Z(i)} \exp\left(-\frac{\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|_{2,a}^2}{h^2}\right)$$

- where $Z(i)$ is the normalizing constant:

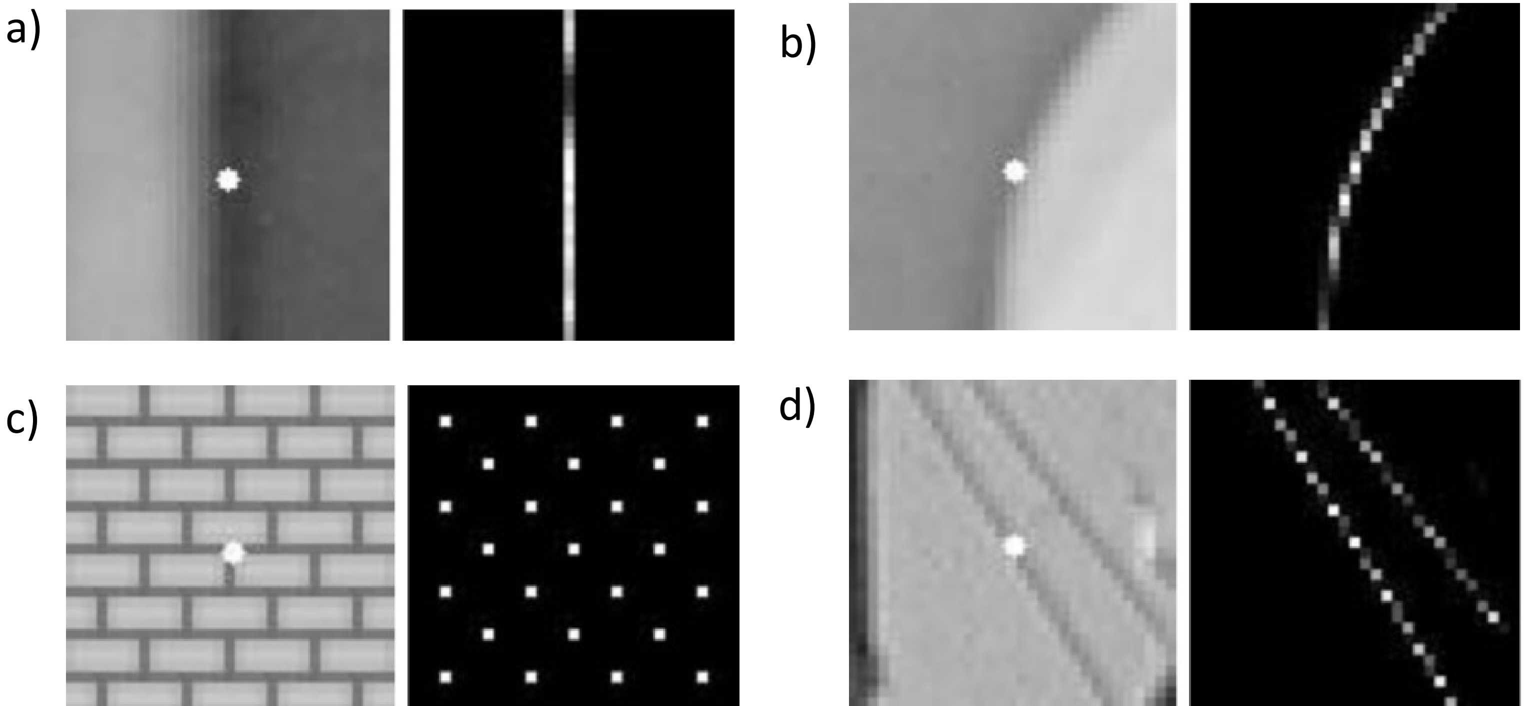
  $Z(i) = \sum_j \exp\left(-\frac{\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|_{2,a}^2}{h^2}\right)$ and the parameter h acts as a degree of filtering, controlling the decay of the exponential function and therefore the decay of weights as a function of Euclidean distances
- NL-means not only compares the gray level in a single point, but the geometrical configuration in the whole neighborhood.
- This fact allows a more robust comparison than neighborhood filters
  - Example: pixel $q3$ has same gray level as pixel p, but neighborhoods are very different and therefore the weight $w(p, q3)$ is nearly zero

a)



Non-local means

- Example: display of the NL-means weight distribution used to estimate the central pixel of every image. Weight range from 1 (white) to 0 (black)
- Shows how NL-means chooses a weighting configuration adapted to local and non-local geometry of the image
- Due to fast decay of exponential kernel, large Euclidean distances lead to nearly zero weights, acting as an automatic threshold
- Most favorable case for NL means is the textured or periodic case, where for every pixel i we can find a very similar configuration (see example c). Natural images have enough redundancy to be restored by NL-means



Computational complexity of NL-means:

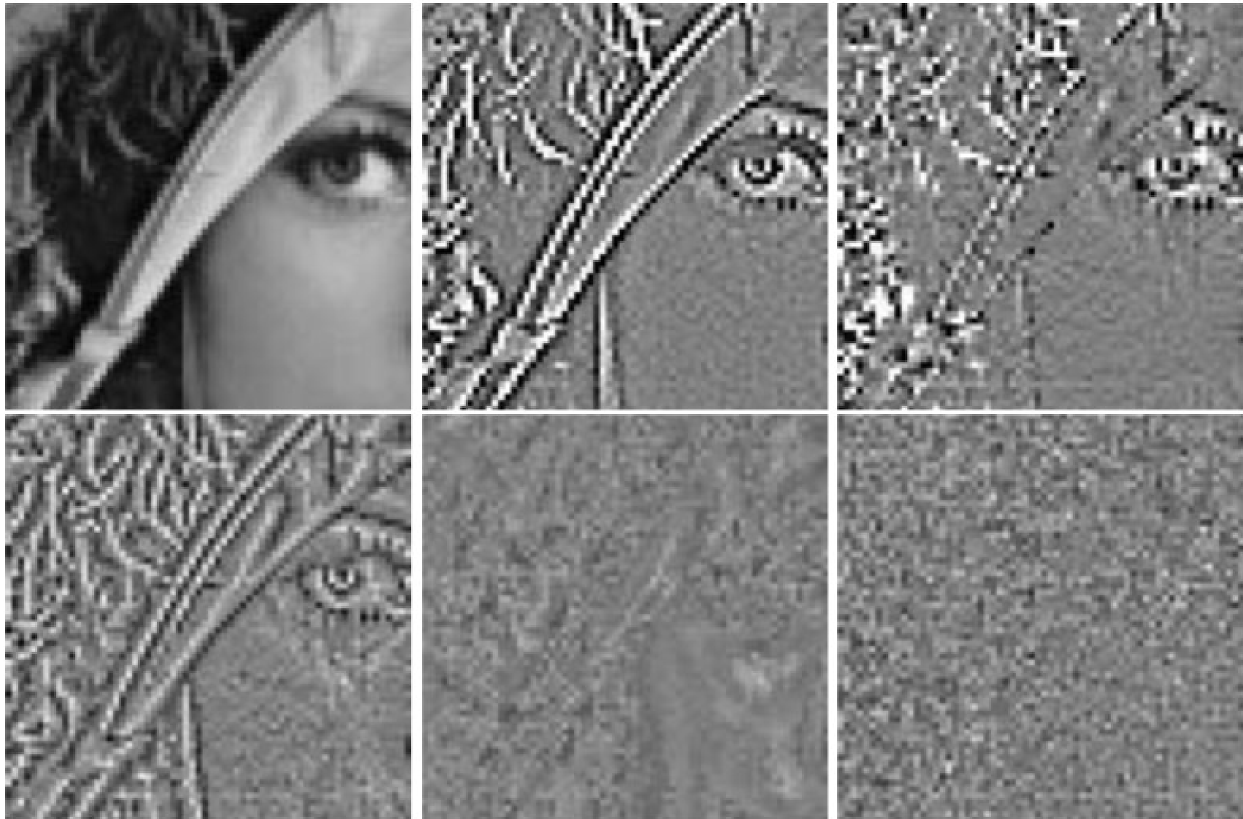
- Restrict the search for similar patches to a 'search window' of $S \times S$ pixels
- Use a similarity neighborhood of $\mathcal{N}_i$ of 7×7 pixels
- If N^2 is number of pixels in the image, final complexity is $49 \cdot S^2 \cdot N^2$

Method noise

- How to know if denoising method was successful?
- Subtract denoised image from original image.
- a) from left to right, top to bottom: noisy image, gaussian filtering, anisotropic filtering, total variation, neighborhood filtering, and NL-means algorithm
- b) image difference (noisy minus denoised), top left is original image

a)

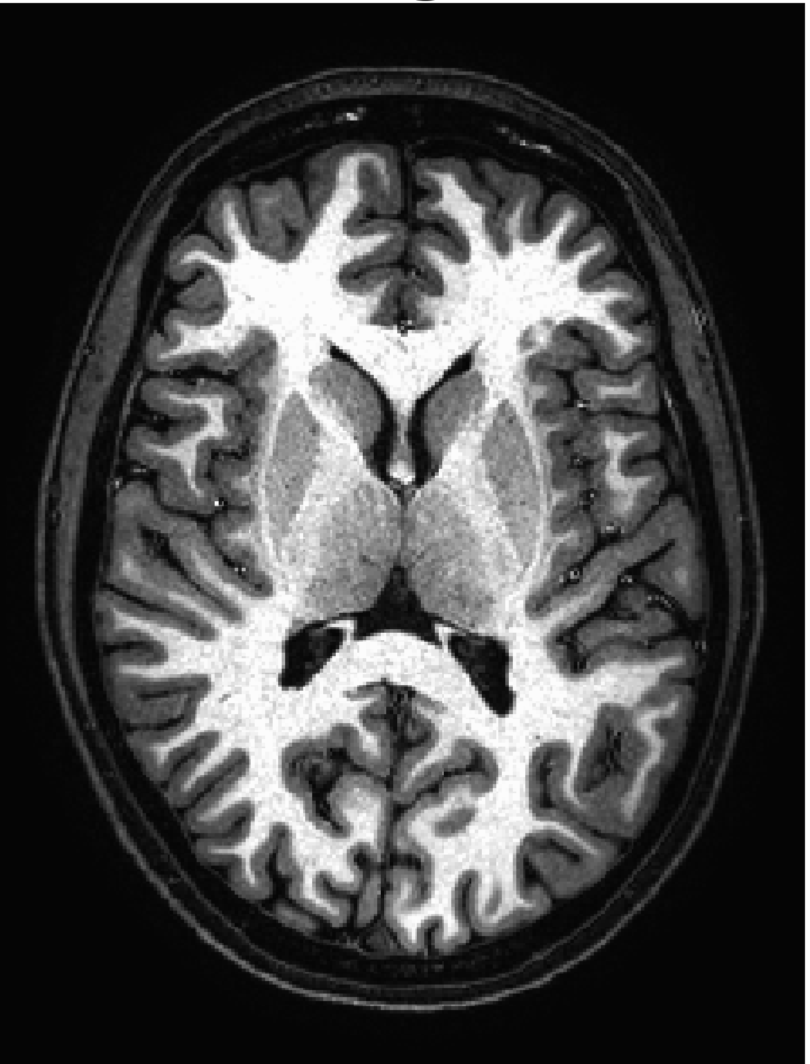
b)



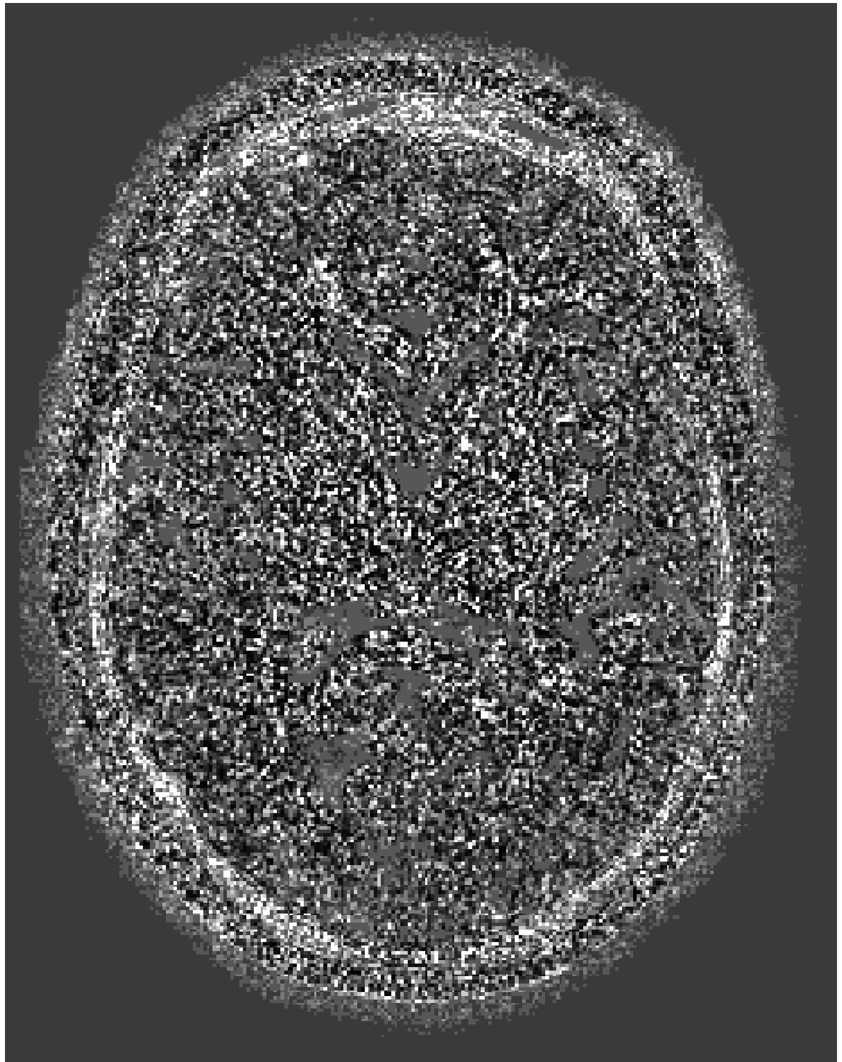
Non-local means on T1 (σ=3)

Image f looks clean, and details are preserved

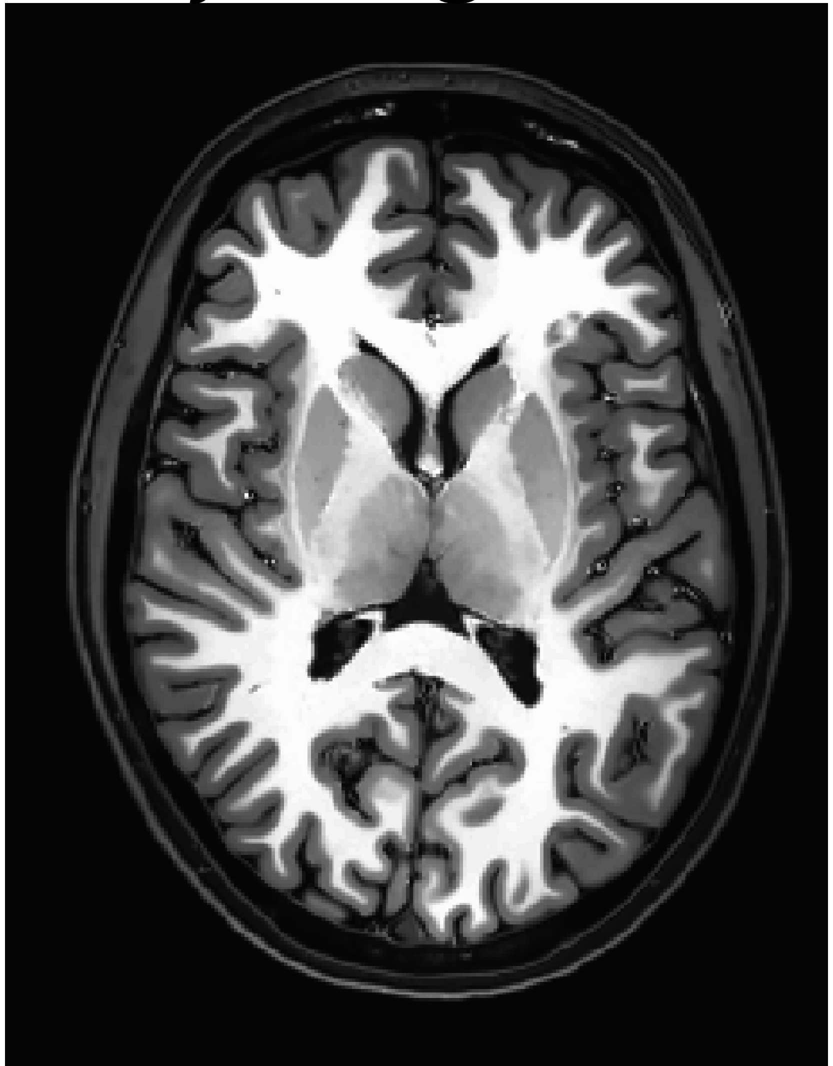
g



n



$f = g - n$



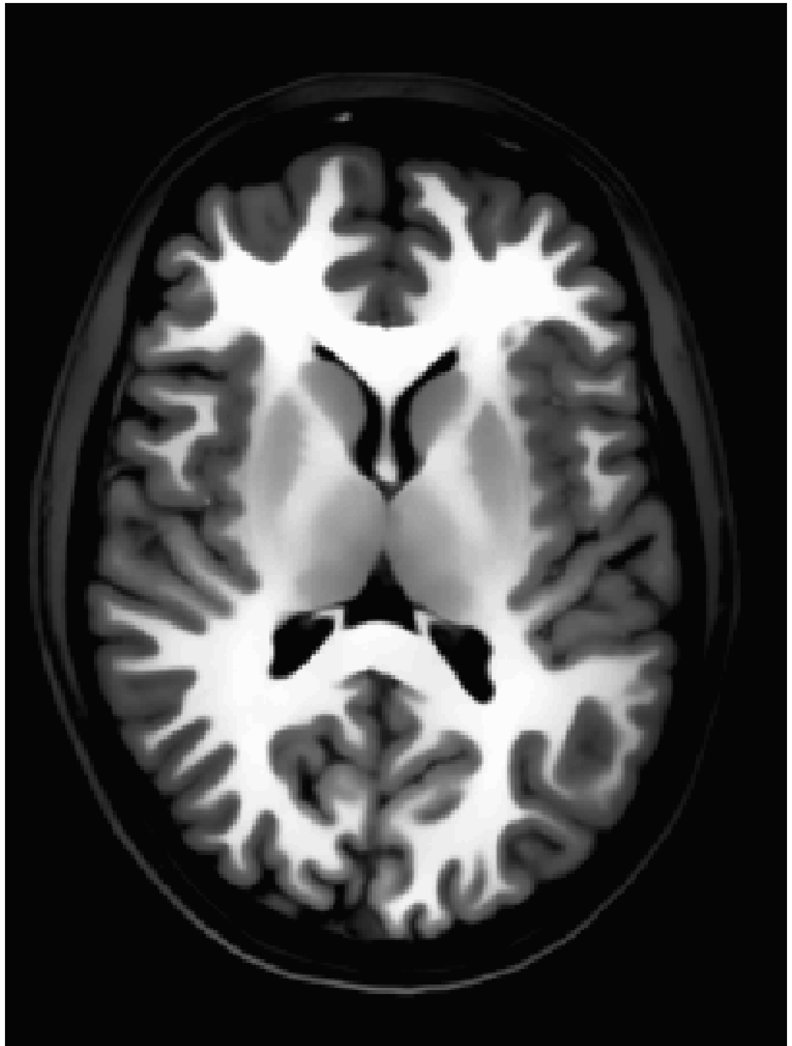
Non-local means on T1 (σ=12)

Image f looks clean, but many details removed! (bad)

g

n

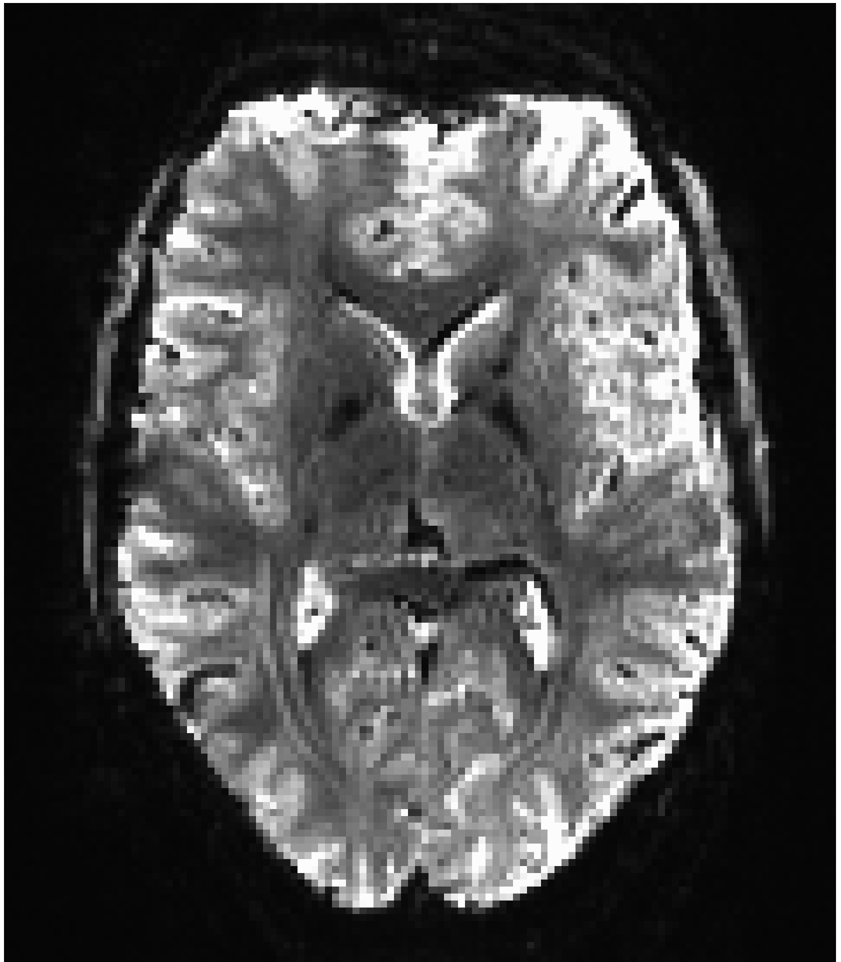
$f = g - n$



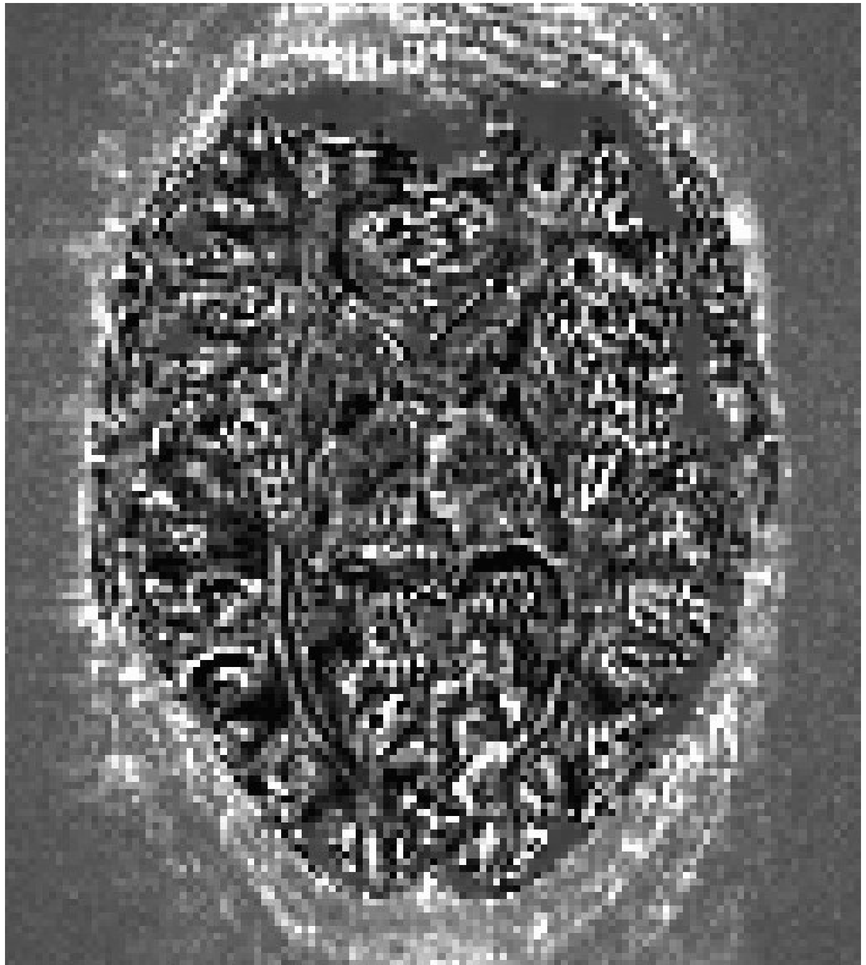
Non-local means on BOLD

Image f looks clean, and details are preserved
However, typically local smoothing is used for BOLD, or a surface-based smoothing analysis

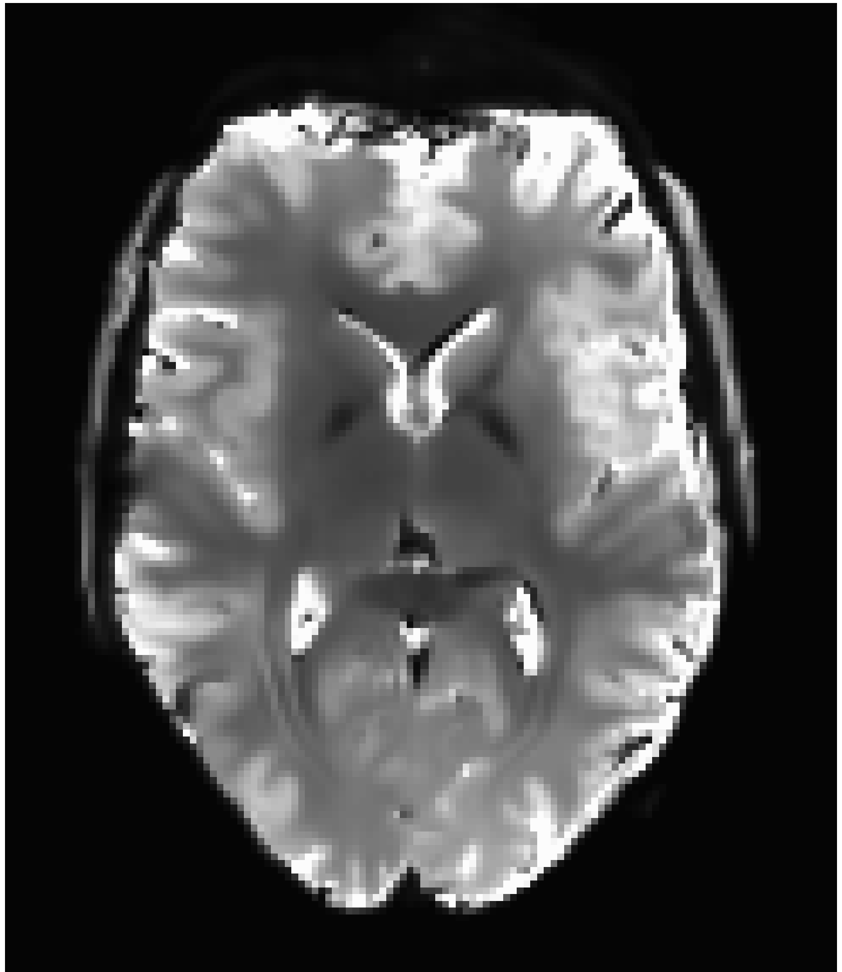
g



n



$f = g - n$

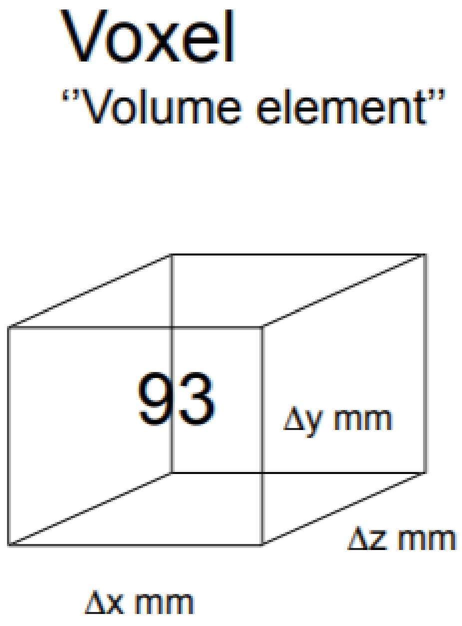


NL-means in dipy

- First get the standard deviation of the noise:
- `sigma = estimate_sigma(img, N=16)`
  - `N=16` Refers to the number of coils in the receiver array of the MRI scanner
- Basically, sigma is the standard deviation of background pixels
- Using sigma can then run nlmeans denoising

```
from dipy.denoise.nlmeans import nlmeans_3d, nlmeans
import nibabel as nib
import numpy as np
import matplotlib.pyplot as plt
from dipy.denoise.noise_estimate import estimate_sigma

tof  = nib.load('C:/shared/swi_tof/sub-19/anats/t1.nii')
img = tof.get_data()
sigma = estimate_sigma(img, N=16)

den = nlmeans(img, sigma)
noise = img - den;

nimg = nib.Nifti1Image(noise, tof.affine)
nib.save(nimg,'C:/shared/swi_tof/sub-19/anats/t1_noise_sigma3.nii.gz')

nimg = nib.Nifti1Image(den, tof.affine)
nib.save(nimg,'C:/shared/swi_tof/sub-19/anats/t1_den_sigma3.nii.gz')
```

Why is noise reduction important in MRI?

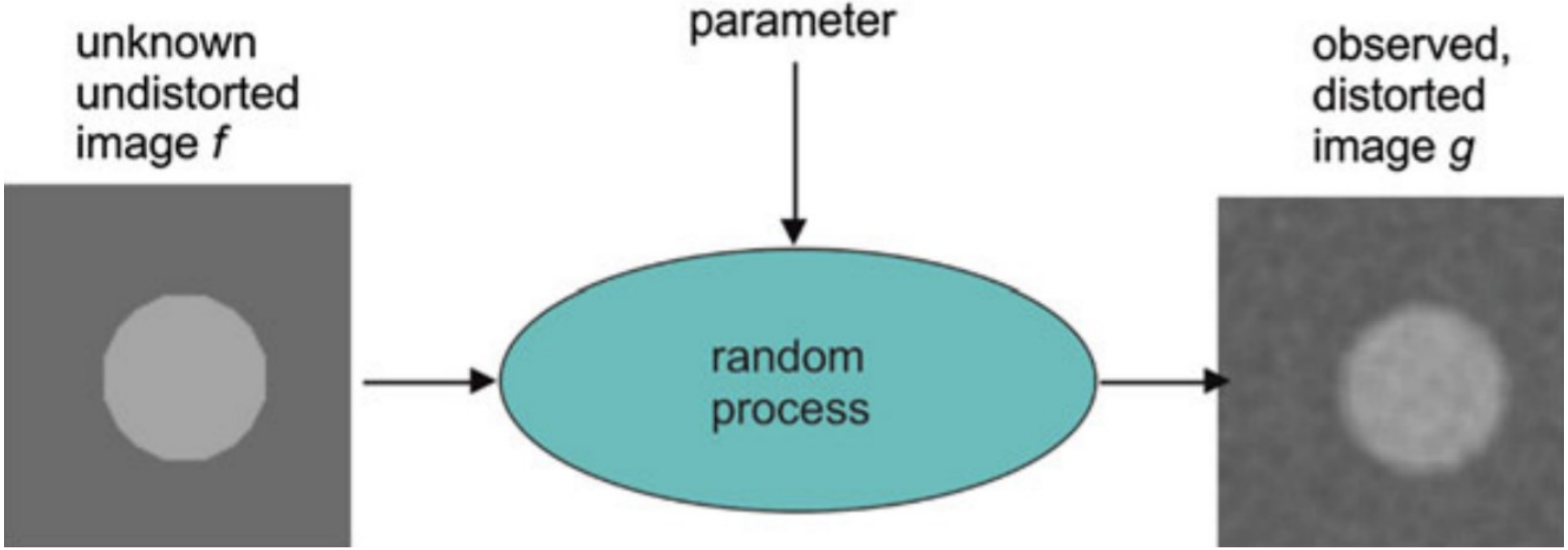


- Consider the voxel:
- The signal we measure coming from the voxel in MRI is proportional to the number of protons in the voxel
- Larger voxels = more protons = stronger signal
- However, we also want high resolution! (smaller voxels)
  - Increasing the resolution decreases the signal coming from each voxel
  - Example 1 : 3x3x3 mm isotropic = 27 cubic millimeters of tissue
  - Example 2: 1x1x1 mm isotropic = 1 cubic millimeter of tissue
- Relationship between SNR is direct and linear. A voxel twice the size of another voxel will have double the SNR
  - In the above case, 3x3x3 mm isotropic voxels have 27 times the SNR!
- Goal is to acquire high-resolution images and then increase SNR by denoising

Bayesian image restoration

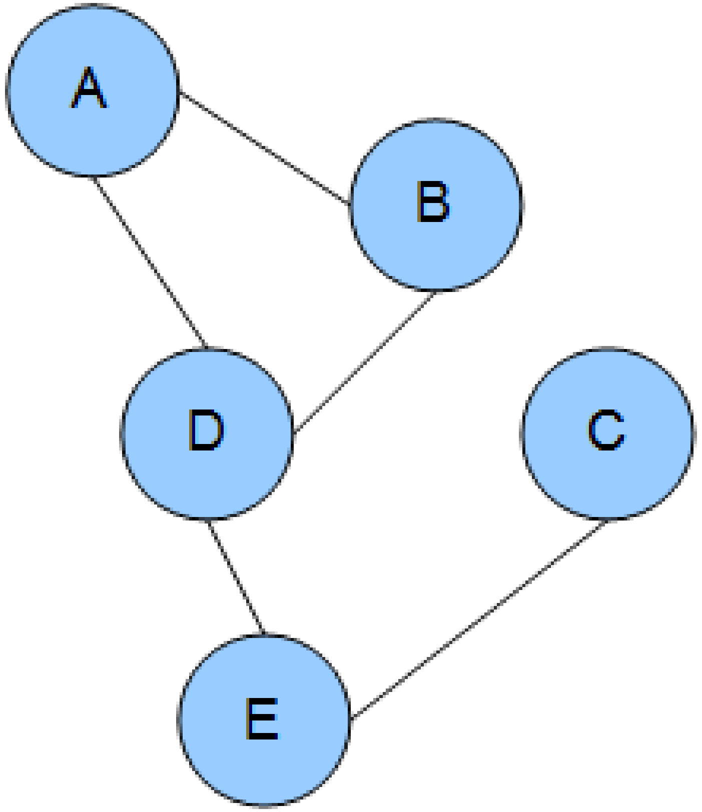
- Linear filtering assumes prior knowledge about the noise, but nothing about the image
- Restoration in a probabilistic framework is able to incorporate smoothness constraints into the method
  - For this purpose, image characteristics are assumed to be representable by a Markov random field
- Determining unknown image function f from some observation g given some probabilistic knowledge about mapping of f to g is a powerful tool based on a simple concept:

Figure: image acquisition as a random process. The aim is to find the most probable undistorted image, given the observation and the parameter of the a-priori knowledge



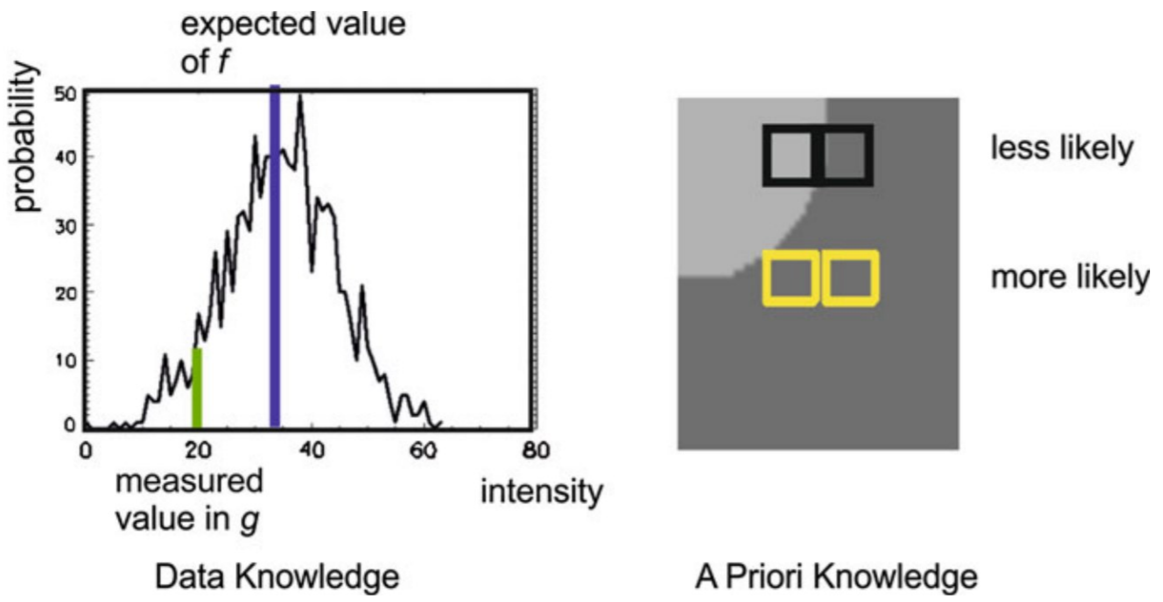
Markov random field

- A Markov random field is a set of random variables having a Markov property and described by an undirected graph
- a) example of a Markov random field, each edge represents a dependency:
  - A depends on B and D
  - B depends on A and D
  - D depends on A, B, and D
  - E depends on D and C
  - C depends on E
- Clique factorization



Bayesian image restoration

- Assume pixels of image are ordered as a vector
- Vector $\boldsymbol{f}$ shall be restored from observed vector $\boldsymbol{g}$
- Individual pixels in the vector will be represented by f_i and g_i respectively
- Noise reduction in Bayesian framework searches for image $\boldsymbol{f}$ that maximizes the conditional probability of observing a noisy image $\boldsymbol{g}$, given that the true image is $\boldsymbol{f}$.
- In Bayesian notation: $P(\boldsymbol{f}|\boldsymbol{g}) \propto P(\boldsymbol{g}|\boldsymbol{f}) \cdot P(\boldsymbol{f})$
- Term $P(\boldsymbol{g}|\boldsymbol{f})$ is the *data term* since it describes the dependency of the observation on the unknown undistorted data $\boldsymbol{f}$
- The term $P(\boldsymbol{f})$ comprises domain knowledge about $\boldsymbol{f}$ independent of an observation (e.g. that in most images, neighboring scene elements have similar values) (b)

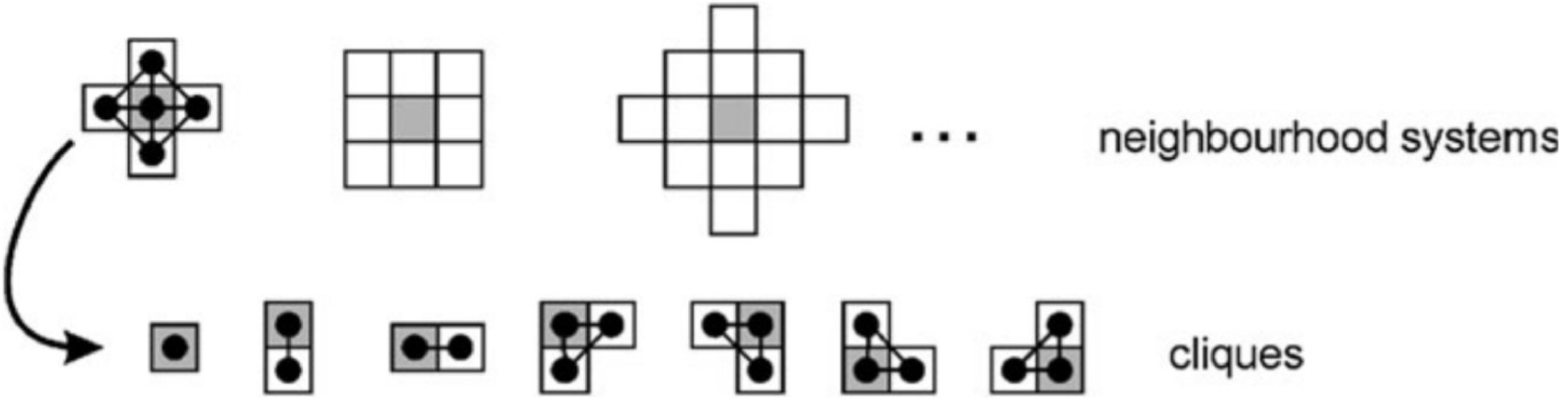


Bayesian image restoration

- Conditional probability of observing $\boldsymbol{g}$ given $\boldsymbol{f}$ depends on type of noise
- We will assume zero-mean Gaussian noise with covariance matrix Σ:
- $P(\boldsymbol{g}|\boldsymbol{f}) = \frac{1}{Z_1}\exp(-\frac{1}{2}\left((\boldsymbol{f}-\boldsymbol{g})^T\Sigma(\boldsymbol{f}-\boldsymbol{g})\right))$
  - Where Z_1 is normalizing constant of the Gaussian distribution
- Assuming probabilities of individual pixels are independent of each other, and they all have the same variance σ^2, the above simplifies to:
- $P(\boldsymbol{g}|\boldsymbol{f}) = \frac{1}{Z_1}\exp(-\sum_{i=0}^{N-1}\frac{(f_i-g_i)^q}{2\sigma^2})$
- The prior $P(\boldsymbol{f})$ is modeled as an MRF among elements of the image $\boldsymbol{f}$
- The probability of an MRF is $P(\boldsymbol{f}) = \frac{1}{Z_2}\exp\left(-U(\omega)\right)$
  - Where Z_2 is a normalization constant and $U(\omega)$ is the clique potential of all cliques in a neighborhood system
  - Neighborhood is defined as spatial adjacency, cliques are possible configurations of scene elements in the neighborhood (next slide)

Bayesian image restoration

- Prior probability for some location in a grid depends on values in freely definable neighborhood
- Given a neighborhood, cliques are configurations of a subset of its elements
- Clique potentials are designed such that they enforce smoothness in the image
- Example of clique energies: $U(\omega) = \beta \sum_{k=0}^{K-1} u_{ik}$, where K is the number of cliques and u_k is the number of pixels in clique k having same intensity as center pixel i



Bayesian image restoration

- $P(\boldsymbol{f}|\boldsymbol{g}) \propto \exp(-\frac{1}{2}(\boldsymbol{f}-\boldsymbol{g})\Sigma(\mathbf{f}-\mathbf{g})^{\mathrm{T}}) \cdot \exp(-U(\omega))$
- $= \prod_{i=0}^{N-1} \exp(-\frac{(f_i-g_i)^2}{2\sigma^2}) \cdot \exp(-\beta \sum_{k=0}^{K-1} u_{i,k})$
- $= \prod_{i=0}^{N-1} \exp(-\frac{(f_i-g_i)^2}{2\sigma^2} - \beta \sum_{k=0}^{K-1} u_{i,k})$
- $= \exp(-\sum_{i=0}^{N-1}(\frac{(f_i-g_i)^2}{2\sigma^2} - \beta \sum_{k=0}^{K-1} u_{i,k}))$
- Maximization of $P(\boldsymbol{f}|\boldsymbol{g})$ means minimizing the exponent:
- $\sum_{i=0}^{N-1}(\frac{(f_i-g_i)^2}{2\sigma^2} - \beta \sum_{k=0}^{K-1} u_{i,k})$
- Finding optimal estimate for $\boldsymbol{f}$ given observation $\boldsymbol{g}$ is difficult, as the number of configurations increases exponentially.
  - Various techniques such as simulated annealing or mean-field annealing exist.