Image Enhancement and Preprocessing

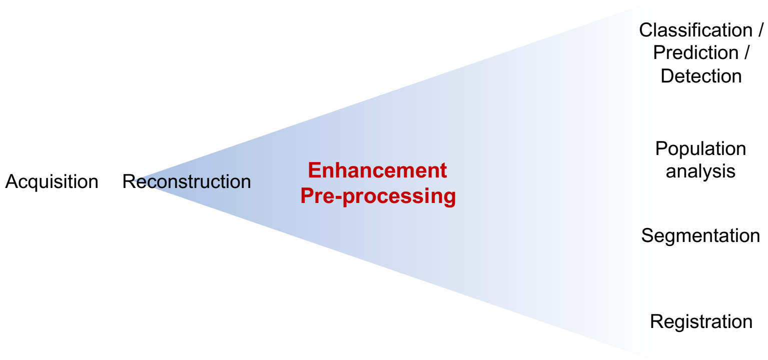
Ender Konukoglu

ETH Zürich

February 25, 2020

Section 1

Introduction



Enhancement and pre-processing tasks form initial steps for any type of analysis. Same applies to methods that use machine learning, probabilistic, variational or energy-based formulations.

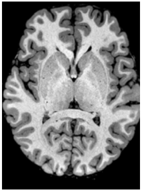
Image enhancement and pre-processing

Enhancement of four conditions

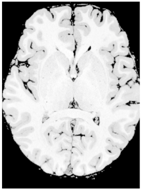
- Bad contrast
- Varying intensity statistics
- Noise
- Bias field
- Variation in pixel-size

Why enhancement?

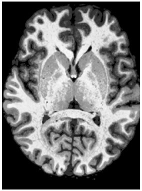
- simplifying interpretation
- better visualization
- normalization for further processing



Original Image

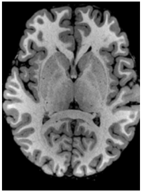


Bad contrast



Intensity variation

Noise



Bias field

Different pixel-size

Outline

- Contrast enhancement
- Intensity normalization
- Noise suppression
- Exercise / Challenge
- Bias correction
- Variations in image and pixel-size

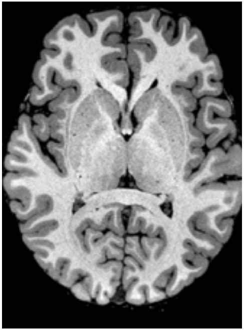
Section 2

Contrast enhancement

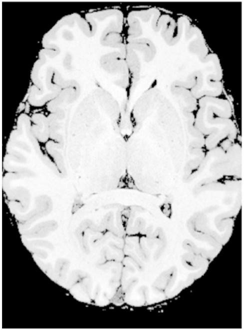
Outline

- Contrast enhancement
  - Gamma correction
  - **Histogram equalization**
- Intensity normalization
- Noise suppression
- Exercise / Challenge
- Bias correction
- Variations in image and pixel-size

Problem description



Original image



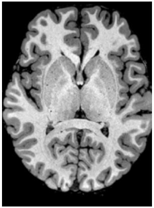
Observation with bad contrast

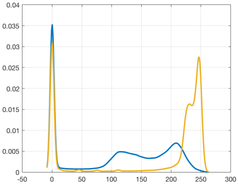
Under-"exposure" or over-"exposure"

How to enhance the contrast of the observation?

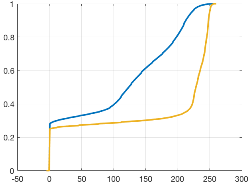
Problem description in terms of histograms

In a nice image intensities are well spread across the range.





Histograms

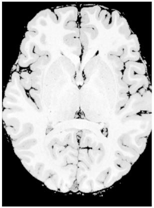


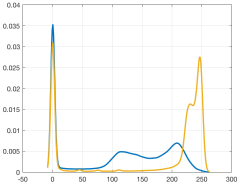
Cumulative histograms

Figure: Blue curves are from the image with good contrast.

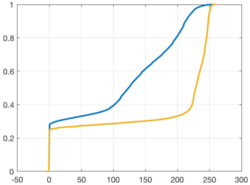
Problem description in terms of histograms

In an image with bad contrast, interesting parts of the image are squeezed in a small intensity range.





Histograms

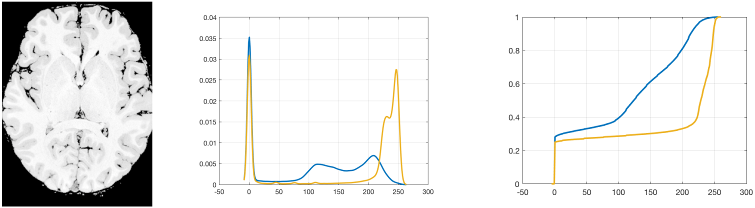


Cumulative histograms

Figure: Yellow curves are from the image with bad contrast.

Problem description in terms of histograms

In an image with bad contrast, interesting parts of the image are squeezed in a small intensity range.



Histograms

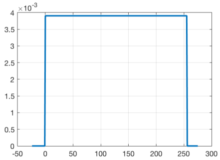
Cumulative histograms

Figure: Yellow curves are from the image with bad contrast.

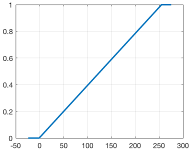
The goal is to distribute the intensities to a larger range

Flat histogram

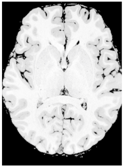
Approach: Flatten the histogram to cover as much intensity region as possible



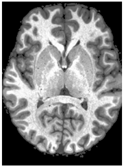
Flat Histogram



CDF of a Flat Histogram



Initial image



Flattened image

Histogram equalization principle

Strategy: Find a **monotonically increasing** intensity mapping that will redistribute intensities to flatten the histogram:

$$\forall i \in [0, i_{max}],\ \ i \to f(i),\ \text{such that} \ \forall\ i \geq j \text{ we have } f(i) \geq f(j)$$

Histogram equalization principle

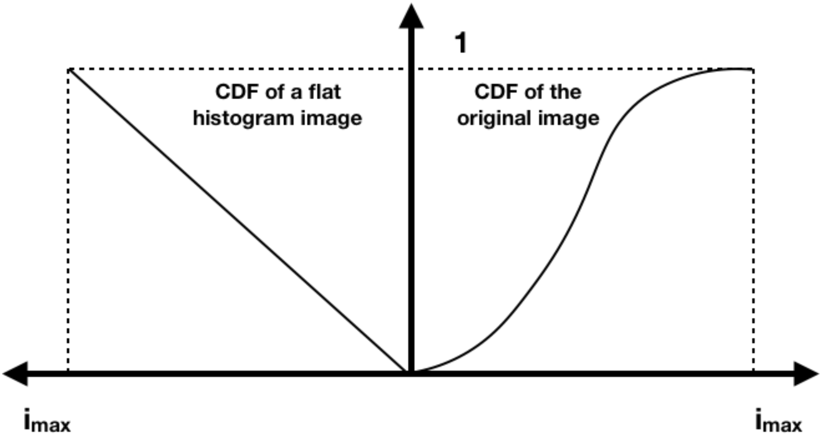
Strategy: Find a **monotonically increasing** intensity mapping that will redistribute intensities to flatten the histogram:

$$\forall i \in [0, i_{max}],\ \ i \rightarrow f(i),\ \text{ such that } \forall\ i \geq j \text{ we have } f(i) \geq f(j)$$

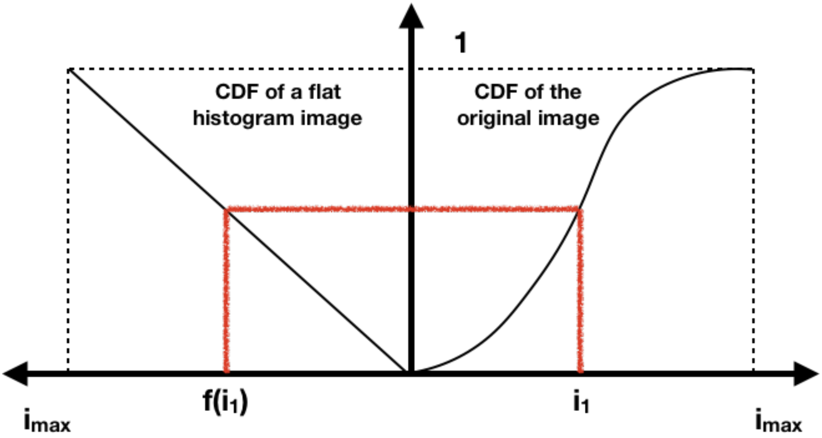
A simple mapping that would achieve this in the continuous domain is the cumulative density distribution itself

$$f(i) = i_{max} P(i) = i_{max} \int_0^i p(j) dj$$

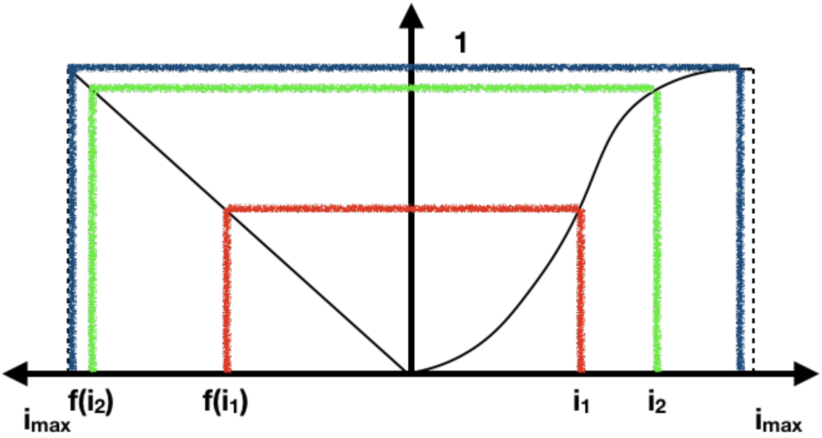
Histogram equalization pictorially



Histogram equalization pictorially



Histogram equalization pictorially



Histogram equalization principle

Strategy: Find a monotonically increasing intensity mapping that will redistribute intensities to flatten the histogram

$$\forall i \in [0, i_{max}],\ i \rightarrow f(i),\ \text{such that } \forall\ i \geq j \text{ we have } f(i) \geq f(j)$$

A simple mapping that would achieve this in the continuous domain is the cumulative density distribution itself

$$f(i) = i_{max} P(i) = i_{max} \int_0^i p(j)dj$$

Histogram equalization principle

Strategy: Find a monotonically increasing intensity mapping that will redistribute intensities to flatten the histogram

$$\forall i \in [0, i_{max}],\ \ i \rightarrow f(i),\ \text{such that}\ \forall\ i \geq j\ \text{we have}\ f(i) \geq f(j)$$

A simple mapping that would achieve this in the continuous domain is the cumulative density distribution itself

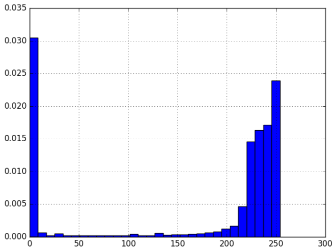
$$f(i) = i_{max} P(i) = i_{max} \int_0^i p(j)dj$$

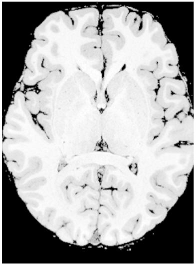
It is easy to see this mapping flattens the distribution

$$p\left(f(i)\right) = p(i)\frac{di}{df(i)} = p(i)\frac{1}{p(i)}\frac{1}{i_{max}} = \frac{1}{i_{max}}$$

Histogram equalization algorithmically

In the quantized image we will get discrete histogram:





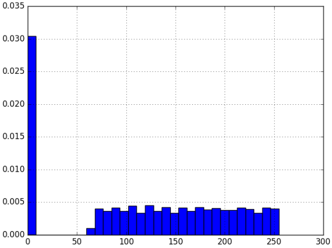
The mapping is:

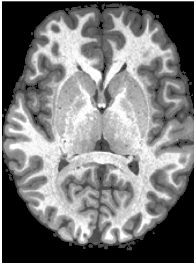
$$f(i) = i_{max} \sum_{0}^{i} h_I(j), \;\; i_{max} : \text{Maximum intensity in the original image.}$$

$h_I(j)$ is the histogram value at intensity j

Histogram equalization result

Corrected histogram and the image:



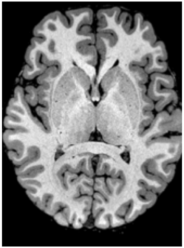


The mapping is then:

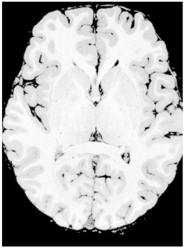
$$f(i) = i_{max} \sum_{0}^{i} h_I(j), \;\; i_{max} : \text{Maximum intensity in the original image.}$$

$h_I(j)$ is the histogram value at intensity j

Question



Original image



Observation with bad contrast

Histogram equalization improves contrast for under and over exposed images.
Can it improve the image if intensities of two different tissue types overlap?

Section 3

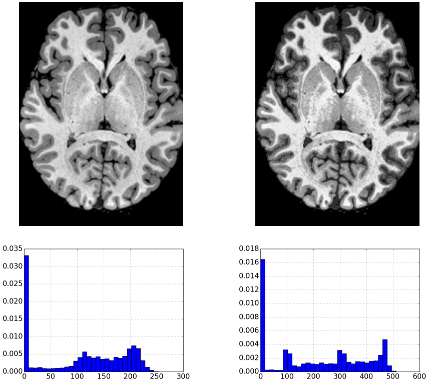
Intensity normalization

Outline

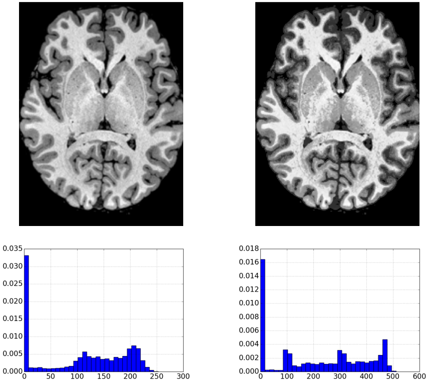
- Contrast enhancement
- Intensity normalization
  - Problem source
  - Basic normalizations
  - Histogram normalization with landmarks
- Noise suppression
- Exercise / Challenge
- Bias correction
- Variations in image and pixel-size

Problem source

- Intensity variations between images

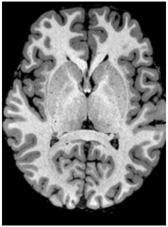


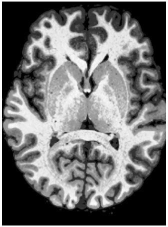
Problem source

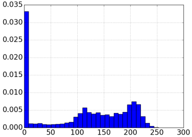


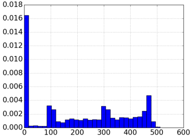
- Intensity variations between images
- Differences in
  - Scanners
  - Protocols
  - Acquisition software
  - Reconstruction method
  - ...

Problem source



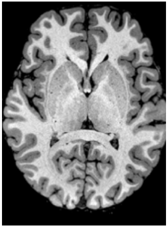


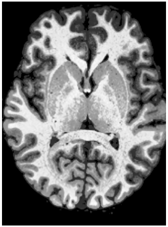


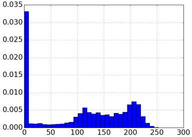


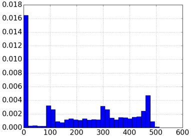
- Intensity variations between images
- Differences in
  - Scanners
  - Protocols
  - Acquisition software
  - Reconstruction method
  - ...
- Particularly important for MRI due to lack of absolute intensities

Problem source



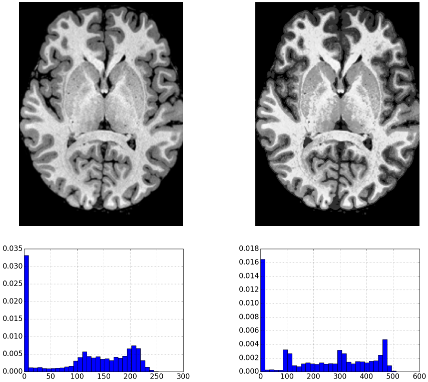






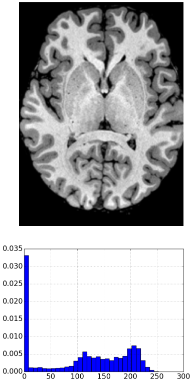
- Intensity variations between images
- Differences in
  - Scanners
  - Protocols
  - Acquisition software
  - Reconstruction method
  - ...
- Particularly important for MRI due to lack of absolute intensities
- Less of a problem for modalities with absolute intensities, e.g. Hounsfield Unit in CT, however still affects algorithmic outcomes.

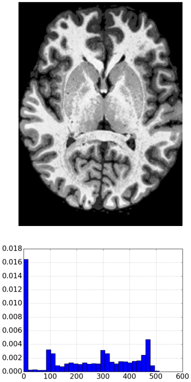
Problem source



- Intensity variations between images
- Differences in
  - Scanners
  - Protocols
  - Acquisition software
  - Reconstruction method
  - ...
- Particularly important for MRI due to lack of absolute intensities
- Less of a problem for modalities with absolute intensities, e.g. Hounsfield Unit in CT, however still affects algorithmic outcomes.
- Adverse effects for the rest of the processing steps

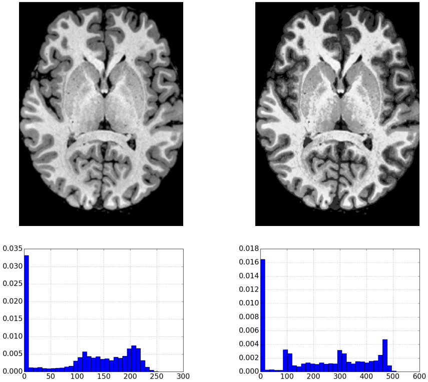
Problem source





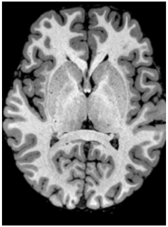
- Intensity variations between images
- Differences in
  - Scanners
  - Protocols
  - Acquisition software
  - Reconstruction method
  - ...
- Particularly important for MRI due to lack of absolute intensities
- Less of a problem for modalities with absolute intensities, e.g. Hounsfield Unit in CT, however still affects algorithmic outcomes.
- Adverse effects for the rest of the processing steps
- Particularly problematic for machine learning methods. Intensity differences between training and test smaples is called domain shift. Think of images at night and day.

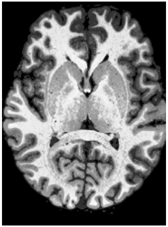
Problem source

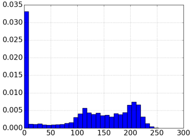


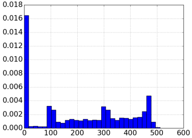
- Intensity variations between images
- Differences in
  - Scanners
  - Protocols
  - Acquisition software
  - Reconstruction method
  - ...
- Particularly important for MRI due to lack of absolute intensities
- Less of a problem for modalities with absolute intensities, e.g. Hounsfield Unit in CT, however still affects algorithmic outcomes.
- Adverse effects for the rest of the processing steps
- Particularly problematic for machine learning methods. Intensity differences between training and test smaples is called domain shift. Think of images at night and day.
- Remains an open problem

In the ideal case









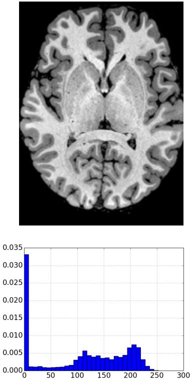
Whenever

- Two images are compared, e.g. registration
- Two groups are compared, e.g. statistical analysis
- A model's parameters are estimated from a set of images, i.e. training, and estimated model is applied on a new set of images, i.e. testing

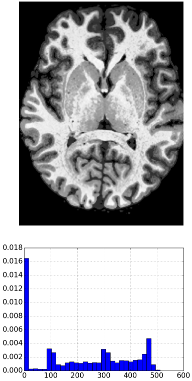
You would like images to have the same intensity "characteristics".

Notion of "characteristics" depends on your definition.

Approach I: Min-max



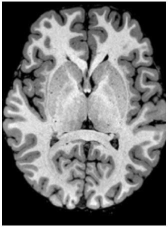
Target Histogram

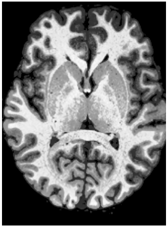


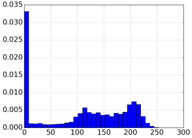
Original Histogram

- Matching the range

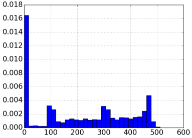
Approach I: Min-max







Target Histogram



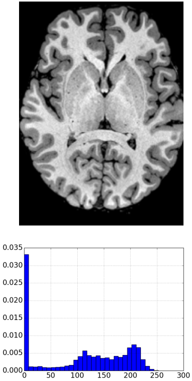
Original Histogram

- Matching the range
- Min-max normalization

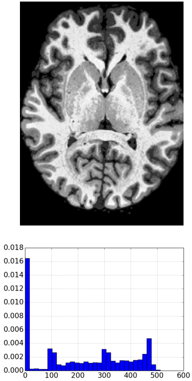
$$f(j) = \frac{j - j_{min}}{j_{max} - j_{min}}(i_{max} - i_{min}) + i_{min}$$

where i and j are the intensities of the two images.

Approach I: Min-max



Target Histogram



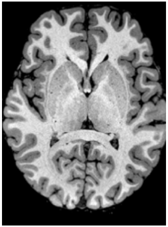
Original Histogram

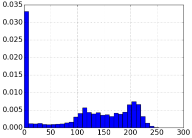
- Matching the range
- Min-max normalization

$$f(j) = \frac{j - j_{min}}{j_{max} - j_{min}}(i_{max} - i_{min}) + i_{min}$$

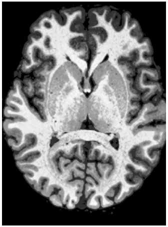
  where i and j are the intensities of the two images.
- max and min values are sensitive to outliers
- Often intensities corresponding to $\epsilon\%$ and $1 - \epsilon\%$ of the CDF are used instead, e.g. $\epsilon = 2$

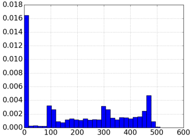
Approach I: Min-max





Target Histogram





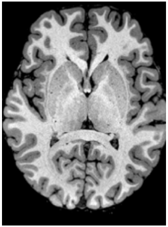
Original Histogram

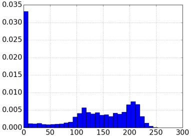
- Matching the range
- Min-max normalization

$$f(j) = \frac{j - j_{min}}{j_{max} - j_{min}}(i_{max} - i_{min}) + i_{min}$$

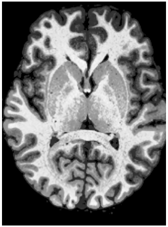
  where i and j are the intensities of the two images.
- max and min values are sensitive to outliers
- Often intensities corresponding to $\epsilon\%$ and $1 - \epsilon\%$ of the CDF are used instead, e.g. $\epsilon = 2$
- Commonly used in machine-learning based methods as a preprocessing step

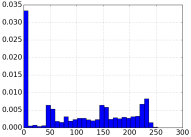
Approach I: Min-max





Target Histogram





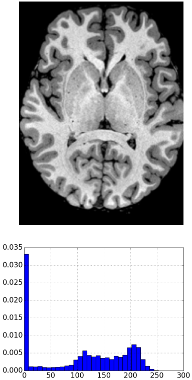
Updated Histogram

- Matching the range
- Min-max normalization

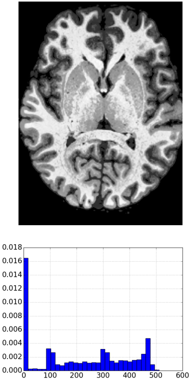
$$f(j) = \frac{j - j_{min}}{j_{max} - j_{min}}(i_{max} - i_{min}) + i_{min}$$

  where i and j are the intensities of the two images.
- max and min values are sensitive to outliers
- Often intensities corresponding to $\epsilon\%$ and $1 - \epsilon\%$ of the CDF are used instead, e.g. $\epsilon = 2$
- Commonly used in machine-learning based methods as a preprocessing step

Approach II: Matching mean and standard deviation



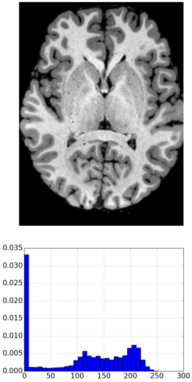
Target Histogram



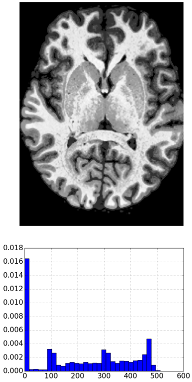
Original Histogram

- Matching second order statistics: Mean and standard deviation normalization

Approach II: Matching mean and standard deviation



Target Histogram



Original Histogram

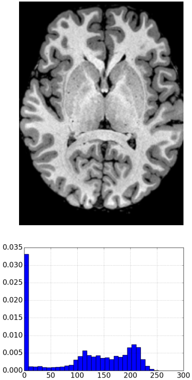
- Matching second order statistics: Mean and standard deviation normalization

$$f(j) = (j - \mu_j)\,\frac{\sigma_i}{\sigma_j} + \mu_i$$

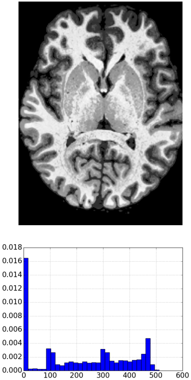
$$\mu_i = \frac{1}{N}\sum_{x=1}^{N} i(x)$$

$$\sigma_i^2 = \frac{1}{N-1}\sum_{x=1}^{N}(i(x) - \mu_i)^2$$

Approach II: Matching mean and standard deviation



Target Histogram



Original Histogram

- Matching second order statistics: Mean and standard deviation normalization

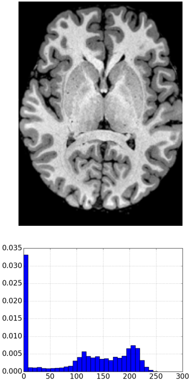
$$f(j) = (j - \mu_j)\frac{\sigma_i}{\sigma_j} + \mu_i$$

$$\mu_i = \frac{1}{N}\sum_{x=1}^{N} i(x)$$

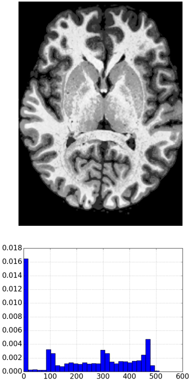
$$\sigma_i^2 = \frac{1}{N-1}\sum_{x=1}^{N}(i(x) - \mu_i)^2$$

- Less sensitive to outliers but still robust statistics may be used for better behavior, e.g. median

Approach II: Matching mean and standard deviation



Target Histogram



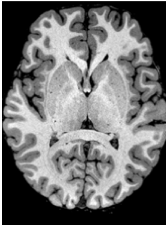
Original Histogram

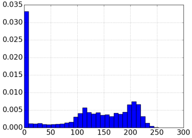
- Matching second order statistics: Mean and standard deviation normalization

$$
\begin{aligned}
f(j) &= (j - \mu_j)\,\frac{\sigma_i}{\sigma_j} + \mu_i \\
\mu_i &= \frac{1}{N}\sum_{x=1}^{N} i(x) \\
\sigma_i^2 &= \frac{1}{N-1}\sum_{x=1}^{N}(i(x) - \mu_i)^2
\end{aligned}
$$

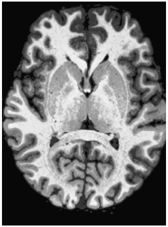
- Less sensitive to outliers but still robust statistics may be used for better behavior, e.g. median
- Often used in machine-learning based methods

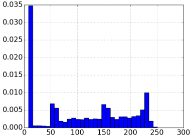
Approach II: Matching mean and standard deviation





Target Histogram





Updated Histogram

- Matching second order statistics: Mean and standard deviation normalization

$$f(j) = (j - \mu_j)\,\frac{\sigma_i}{\sigma_j} + \mu_i$$

$$\mu_i = \frac{1}{N}\sum_{x=1}^{N} i(x)$$

$$\sigma_i^2 = \frac{1}{N-1}\sum_{x=1}^{N}(i(x) - \mu_i)^2$$

- Less sensitive to outliers but still robust statistics may be used for better behavior
- Often used in machine-learning based methods

Question

Min-max matching

$$f(j) = \frac{j - j_{min}}{j_{max} - j_{min}}(i_{max} - i_{min}) + i_{min}$$

Mean - std matching

$$\begin{aligned} f(j) &= (j - \mu_j)\,\frac{\sigma_i}{\sigma_j} + \mu_i \\ \mu_i &= \frac{1}{N}\sum_{x=1}^{N} i(x) \\ \sigma_i^2 &= \frac{1}{N-1}\sum_{x=1}^{N}(i(x) - \mu_i)^2 \end{aligned}$$

Both min-max and mean-std matching are linear mappings. True or False?

Question

Mean - std matching

$$
\begin{aligned}
f(j) &= (j - \mu_j)\,\frac{\sigma_i}{\sigma_j} + \mu_i \\
\mu_i &= \frac{1}{N}\sum_{x=1}^{N} i(x) \\
\sigma_i^2 &= \frac{1}{N-1}\sum_{x=1}^{N}(i(x) - \mu_i)^2
\end{aligned}
$$

Min-max matching

$$f(j) = \frac{j - j_{min}}{j_{max} - j_{min}}(i_{max} - i_{min}) + i_{min}$$

Both min-max and mean-std matching are linear mappings. True or False? True

Can these transformations change the contrast of an image?

Question

Mean - std matching

$$
\begin{aligned}
f(j) &= (j - \mu_j)\,\frac{\sigma_i}{\sigma_j} + \mu_i \\
\mu_i &= \frac{1}{N}\sum_{x=1}^{N} i(x) \\
\sigma_i^2 &= \frac{1}{N-1}\sum_{x=1}^{N} (i(x) - \mu_i)^2
\end{aligned}
$$

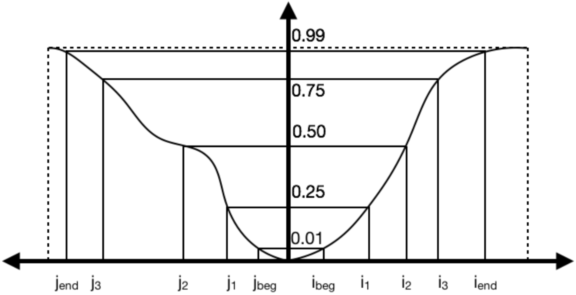
Min-max matching

$$
f(j) = \frac{j - j_{min}}{j_{max} - j_{min}}(i_{max} - i_{min}) + i_{min}
$$

Both min-max and mean-std matching are linear mappings. True or False? True

Can these transformations change the contrast of an image? No

Approach III: Histogram normalization with landmarks



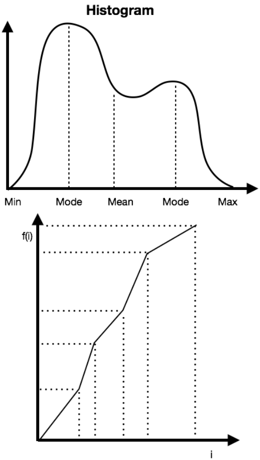
Example: Quartiles and min-max as landmarks

- Piecewise linear mapping

$$f(j) = \begin{cases} (j - j_{beg})\frac{i_1 - i_{beg}}{j_1 - j_{beg}} + i_{beg}; & j \leq j_1 \\ (j - j_1)\frac{i_2 - i_1}{j_2 - j_1} + i_1; & j_1 < j \leq j_2 \\ (j - j_2)\frac{i_3 - i_2}{j_3 - j_2} + i_2; & j_2 < j \leq j_3 \\ (j - j_3)\frac{i_{end} - i_3}{j_{end} - j_3} + i_3; & j_3 < j \end{cases}$$

Histogram normalization: Nyul's method

[Nyul, Udupa and Zhang, IEEE TMI 2000]

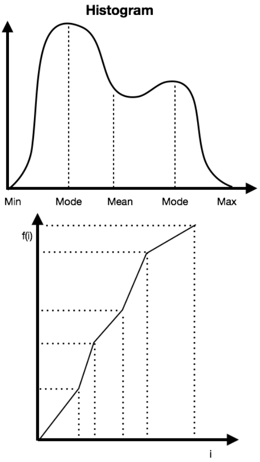


Piecewise linear map

- Matching landmarks of image histograms

Histogram normalization: Nyul's method

[Nyul, Udupa and Zhang, IEEE TMI 2000]

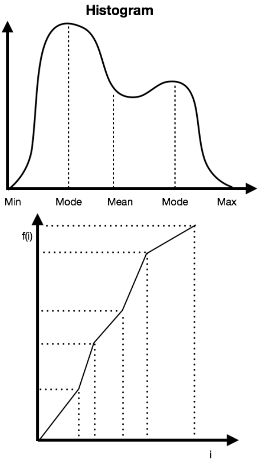


Piecewise linear map

- Matching landmarks of image histograms
- Piece-wise linear map: non-linear mapping

Histogram normalization: Nyul's method

[Nyul, Udupa and Zhang, IEEE TMI 2000]

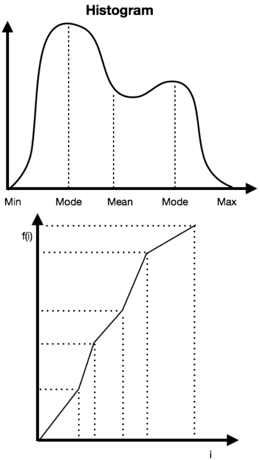


Piecewise linear map

- Matching landmarks of image histograms
- Piece-wise linear map: non-linear mapping
- Different landmarks are possible
  - min and max
  - mean or median
  - quartiles, deciles
  - modes, ...

Histogram normalization: Nyul's method

[Nyul, Udupa and Zhang, IEEE TMI 2000]

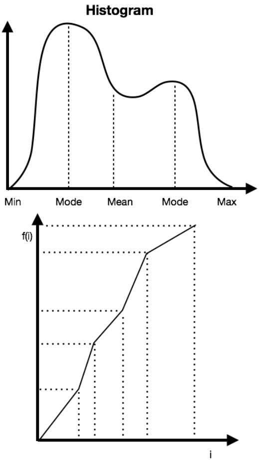


Piecewise linear map

- Matching landmarks of image histograms
- Piece-wise linear map: non-linear mapping
- Different landmarks are possible
  - min and max
  - mean or median
  - quartiles, deciles
  - modes, . . .
- Less sensitive to outliers

Histogram normalization: Nyul's method

[Nyul, Udupa and Zhang, IEEE TMI 2000]

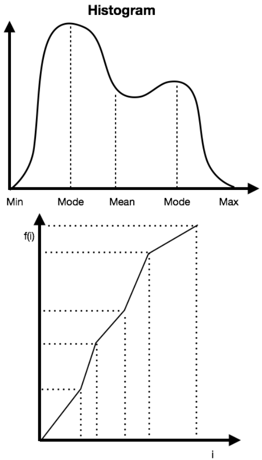


Piecewise linear map

- Matching landmarks of image histograms
- Piece-wise linear map: non-linear mapping
- Different landmarks are possible
  - min and max
  - mean or median
  - quartiles, deciles
  - modes, ...
- Less sensitive to outliers
- Widely used

Histogram normalization: Nyul's method

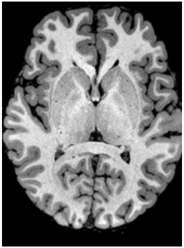
[Nyul, Udupa and Zhang, IEEE TMI 2000]

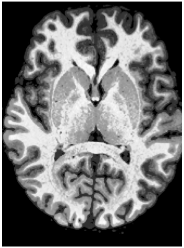


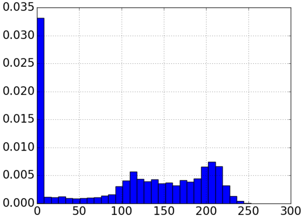
Piecewise linear map

- Matching landmarks of image histograms
- Piece-wise linear map: non-linear mapping
- Different landmarks are possible
  - min and max
  - mean or median
  - quartiles, deciles
  - modes, . . .
- Less sensitive to outliers
- Widely used
- Population-wide use:
  - Estimate landmark positions from population data as averages
  - Map the new image's landmarks to the averages

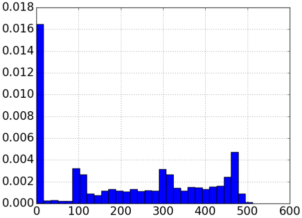
Correction results





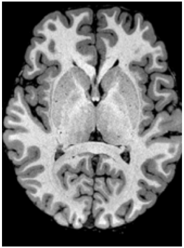


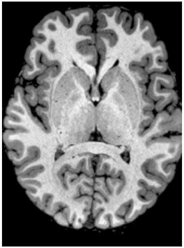
Target Histogram

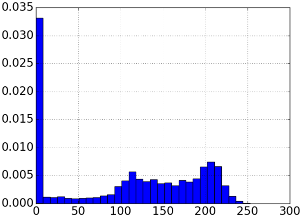


Original Histogram

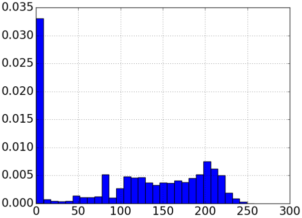
Correction results





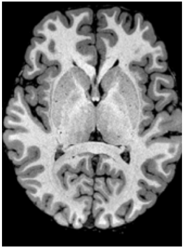


Target Histogram

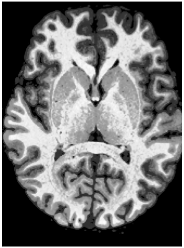


Updated Histogram

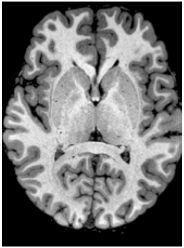
Question



Target image



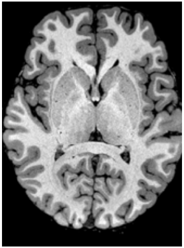
Original image

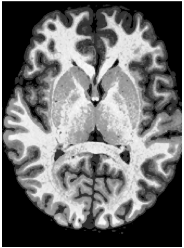


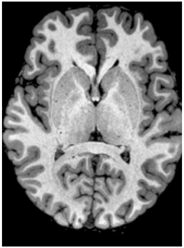
Matched image

Why did the contrast change?

Question


Target image


Original image


Matched image

Why did the contrast change? Due to the non-linearity of the mapping.

Analysis

Several things to pay attention to

Analysis

Several things to pay attention to

- Image content

Analysis

Several things to pay attention to

- Image content
  - Histogram matching does not know about image content
  - All pixels are considered independent
  - No spatial information / correlation between neighboring pixels
  - Matching gradients has been proposed

Analysis

Several things to pay attention to

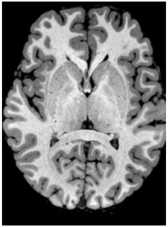
- Image content
  - Histogram matching does not know about image content
  - All pixels are considered independent
  - No spatial information / correlation between neighboring pixels
  - Matching gradients has been proposed
- Object size

Analysis

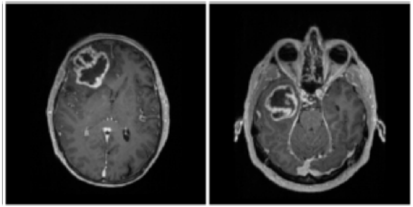
Several things to pay attention to

- Image content
  - Histogram matching does not know about image content
  - All pixels are considered independent
  - No spatial information / correlation between neighboring pixels
  - Matching gradients has been proposed
- Object size
  - Differences in object size can be a problem
  - Perfect histogram match would not be preferred in that case
  - Use *mode* matching if object sizes are different

Analysis - Pathologies



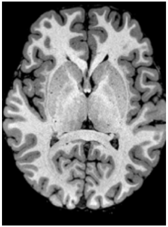
Images of healthy brain



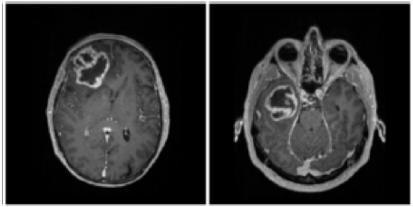
Images with brain tumors

When matching histograms of images with large lesions and healthy brains, should we expect any problems?

Analysis - Pathologies



Images of healthy brain

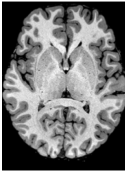


Images with brain tumors

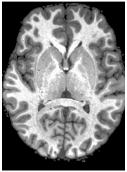
When matching histograms of images with large lesions and healthy brains, should we expect any problems?

- Presence of pathologies will change histograms
- Matching pathology bearing images is hard
- *Robust statistics* and *outlier detection* can help

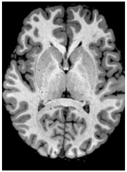
Discussion



Original



Histogram eq.



Nyul's method

What is the relationship between histogram equalization and intensity normalization?

In which circumstances intensity normalization would be preferred?

Exercise

Come up with an image example where histogram equalization would lead to a bad result while mode matching yields the correct result. [Hint: think about object size and multiple objects]

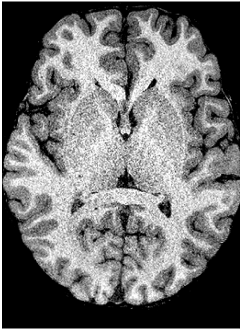
Section 4

Noise suppression

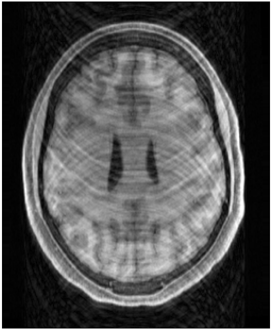
Outline

- Contrast enhancement
- Intensity normalization
- Noise suppression
    - Problem
    - Linear filters
    - Median filters
    - Non-local means
    - Energy-based
- Exercise / Challenge
- Bias correction
- Variations in image and pixel-size

Problem source



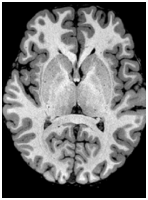
Random noise



Imaging artifacts

- Imperfections in the acquisition device
- Artifacts due to fast acquisition
- Artifacts due to implants
- ...

Problem definition



Clean image: I

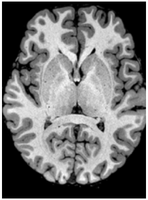
Noisy image: J

- Additive noise

$$J(x) = I(x) + \epsilon(x).$$

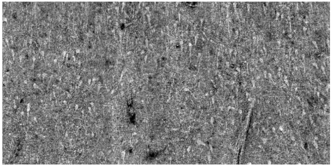
$\epsilon(x)$ is often considered to be iid
Example: Gaussian noise ($\epsilon \sim \mathcal{N}(0, \sigma)$)

Problem definition



Clean image: I

Noisy image: J



Speckle

- Additive noise

$$J(x) = I(x) + \epsilon(x).$$

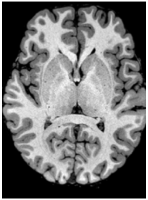
  $\epsilon(x)$ is often considered to be iid
  Example: Gaussian noise ($\epsilon \sim \mathcal{N}(0, \sigma)$)

- Multiplicative noise

$$J(x) = I(x)\epsilon(x)$$

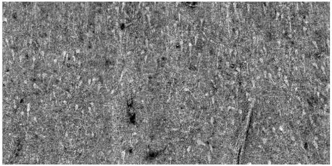
  Example: Speckle in ultrasound and optical coherence tomography

Problem definition



Clean image: I

Noisy image: J



Speckle

- Additive noise

$$J(x) = I(x) + \epsilon(x).$$

  $\epsilon(x)$ is often considered to be iid
  Example: Gaussian noise ($\epsilon \sim \mathcal{N}(0, \sigma)$)

- Multiplicative noise

$$J(x) = I(x)\epsilon(x)$$

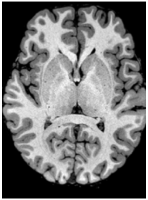
  Example: Speckle in ultrasound and optical coherence tomography

- More complicated, example: Noise in MR magnitude image

$$J(x) = \sqrt{(I_r(x) + \epsilon)^2 + (I_i(x) + \eta)^2}$$

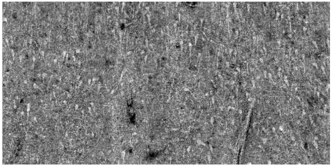
  $\epsilon, \eta \sim \mathcal{N}(0, \sigma)$, Rician noise
  [Gudbjartsson and Patz, MRM 1995]

Problem definition



Clean image: I

Noisy image: J



Speckle

- Additive noise

$$J(x) = I(x) + \epsilon(x).$$

  $\epsilon(x)$ is often considered to be iid
  Example: Gaussian noise ($\epsilon \sim \mathcal{N}(0, \sigma)$)

- Multiplicative noise

$$J(x) = I(x)\epsilon(x)$$

  Example: Speckle in ultrasound and optical coherence tomography

- More complicated, example: Noise in MR magnitude image

$$J(x) = \sqrt{(I_r(x) + \epsilon)^2 + (I_i(x) + \eta)^2}$$

  $\epsilon, \eta \sim \mathcal{N}(0, \sigma)$, Rician noise
  [Gudbjartsson and Patz, MRM 1995]

- Given the noisy image J remove noise to get I

Methods overview

- Generic methods
  - Convolutional / linear
  - Median filtering
  - Anisotropic diffusion
  - Non-local means
  - Optimization based
- Specialized methods: For specific applications and noise models - we will not study them.

Subsection 1

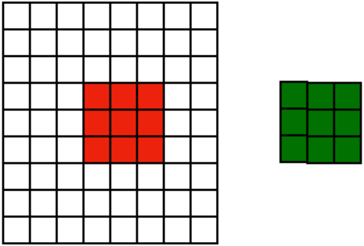
Linear filters

Convolutional / linear filtering

Convolution

$$
\begin{aligned}
\tilde{I} &= J * S \\
\tilde{I}(x,y,z) &= \int_r \int_p \int_q J(x-r, y-p, z-q) S(r,p,q) drdpdq \\
\tilde{I}(i,j,k) &= \sum_r \sum_p \sum_q J(i-r, j-p, k-q) S(r,p,q)
\end{aligned}
$$

S: structural element (kernel) and $\tilde{I}$ is the denoised image

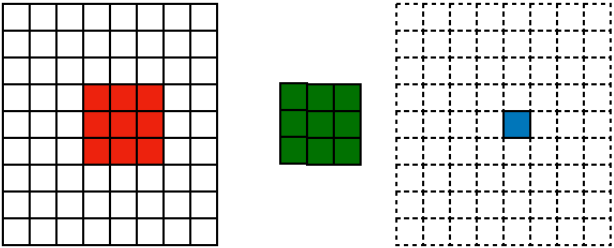


Convolutional / linear filtering

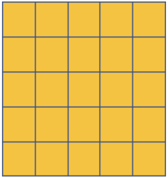
Convolution

$$
\begin{aligned}
\tilde{I} &= J * S \\
\tilde{I}(x, y, z) &= \int_r \int_p \int_q J(x-r, y-p, z-q) S(r, p, q) drdpdq \\
\tilde{I}(i, j, k) &= \sum_r \sum_p \sum_q J(i-r, j-p, k-q) S(r, p, q)
\end{aligned}
$$

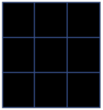
S: structural element (kernel) and $\tilde{I}$ is the denoised image



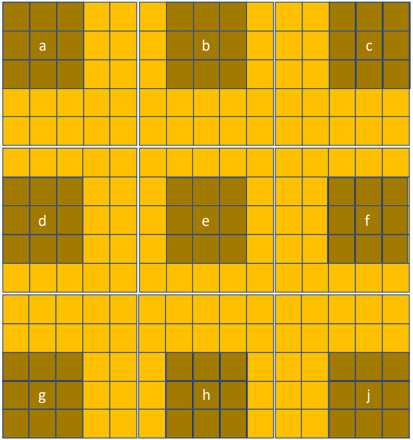
Convolutional / linear filtering - graphically

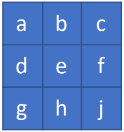


Image



Kernel





Commonly used convolutional filters for noise suppression: Mean filtering

Mean filtering

- average in a neighborhood

$$S_3 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- larger kernels similar, e.g. S_5 and S_7
- larger kernels smooths more

Commonly used convolutional filters for noise suppression: Mean filtering

Mean filtering

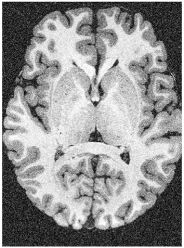
- average in a neighborhood

$$S_3 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- larger kernels similar, e.g. S_5 and S_7
- larger kernels smooths more
- separable: very efficient to implement

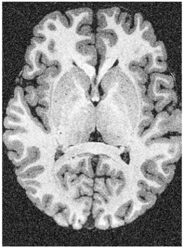
$$S_3 = \frac{1}{9}[1, 1, 1] * \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Mean filtering examples

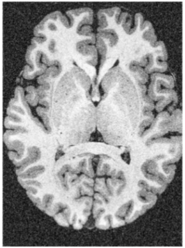


Noisy

Mean filtering examples

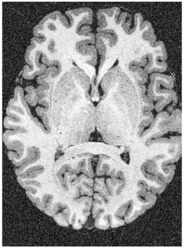


Noisy

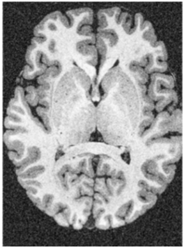


S_3

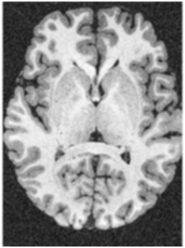
Mean filtering examples



Noisy

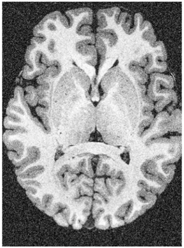


S_3

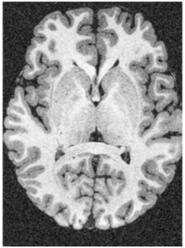


S_5

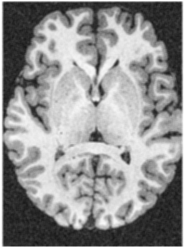
Mean filtering examples



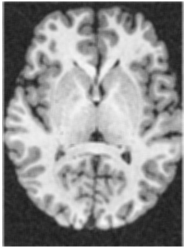
Noisy



S_3



S_5



S_7

Commonly used convolutional filters for noise suppression: Binomial filtering

Binomial filtering

- iterative convolutions of $[1, 1]$
- integer filters

$$S_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Commonly used convolutional filters for noise suppression: Binomial filtering

Binomial filtering

- iterative convolutions of $[1, 1]$
- integer filters

$$S_3 = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- larger kernels by successive convolutions of smaller
- larger kernels smooth more

Commonly used convolutional filters for noise suppression: Binomial filtering

Binomial filtering

- iterative convolutions of $[1, 1]$
- integer filters

$$S_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- larger kernels by successive convolutions of smaller
- larger kernels smooth more
- separable

$$S_3 = \frac{1}{16}[1, 2, 1] * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

Commonly used convolutional filters for noise suppression: Binomial filtering

Binomial filtering

- iterative convolutions of $[1, 1]$
- integer filters

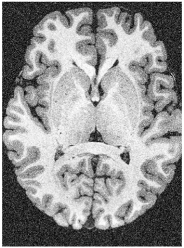
$$S_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- larger kernels by successive convolutions of smaller
- larger kernels smooth more
- separable

$$S_3 = \frac{1}{16} [1, 2, 1] * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

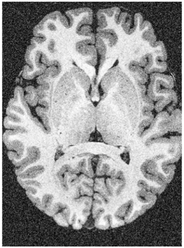
- better properties than averaging filtering in the frequency domain - we will not see this further.

Binomial filtering examples

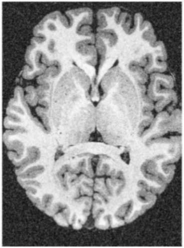


Noisy

Binomial filtering examples

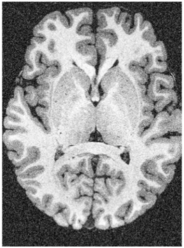


Noisy

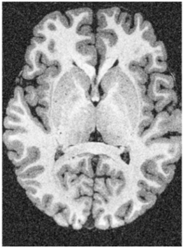


S_3

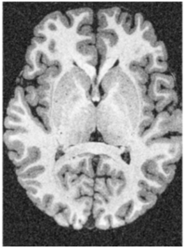
Binomial filtering examples



Noisy

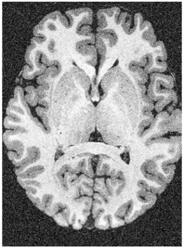


S_3

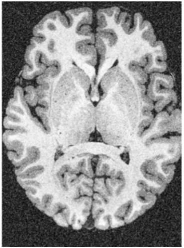


S_5

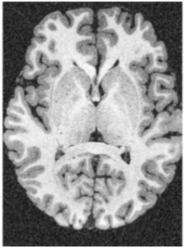
Binomial filtering examples



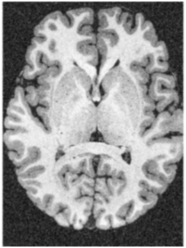
Noisy



S_3



S_5



S_7

Commonly used convolutional filters for noise suppression: Gaussian filtering

Gaussian filtering

- construction by Gaussian kernel centered at the center of the structural element, e.g.

$$S_3(i,j) = \frac{1}{C} \exp\left(-\frac{(i-1)^2 + (j-1)^2}{2\sigma^2}\right), \quad C : \text{Normalizing constant}$$

Commonly used convolutional filters for noise suppression: Gaussian filtering

Gaussian filtering

- construction by Gaussian kernel centered at the center of the structural element, e.g.

$$S_3(i,j) = \frac{1}{C} \exp\left(-\frac{(i-1)^2 + (j-1)^2}{2\sigma^2}\right), \quad C : \text{Normalizing constant}$$

- has infinite support
- finite structural approximates the Gaussian kernel

Commonly used convolutional filters for noise suppression: Gaussian filtering

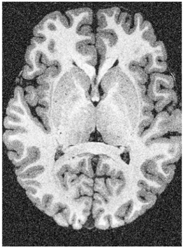
Gaussian filtering

- construction by Gaussian kernel centered at the center of the structural element, e.g.

$$S_3(i,j) = \frac{1}{C}\exp\left(-\frac{(i-1)^2+(j-1)^2}{2\sigma^2}\right), \quad C : \text{Normalizing constant}$$

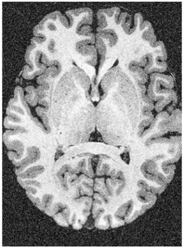
- has infinite support
- finite structural approximates the Gaussian kernel
- structural element size depends on σ, e.g. large σ needs larger size for better approximation
- larger σ filters smooth more

Gaussian filtering examples

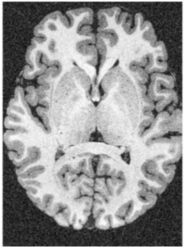


Noisy

Gaussian filtering examples

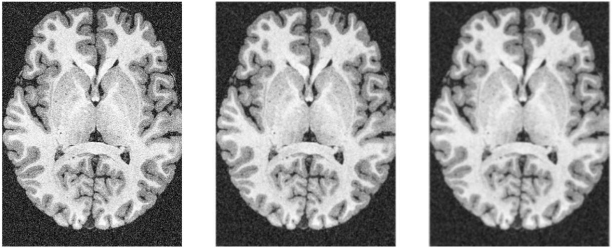


Noisy



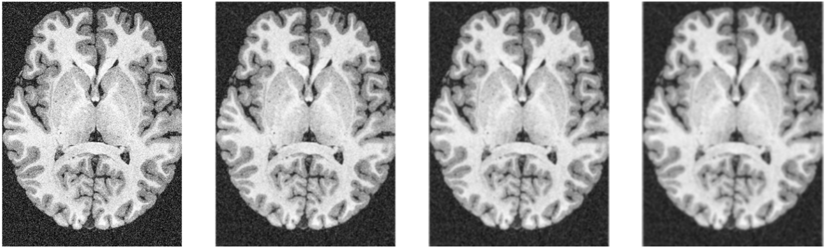
$\sigma = 1$

Gaussian filtering examples



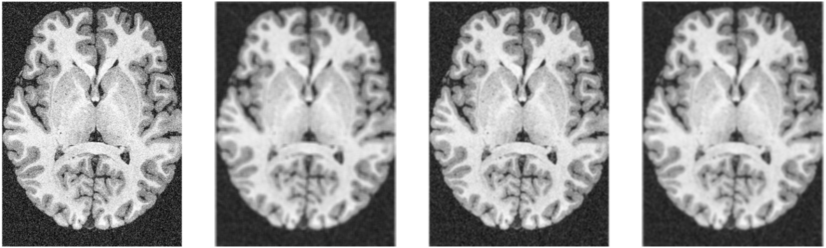
Noisy $\quad \sigma = 1 \quad \sigma = 1.5$

Gaussian filtering examples



Noisy | $\sigma = 1$ | $\sigma = 1.5$ | $\sigma = 2$

Analysis



Noisy | Mean with S_7 | Binom with S_7 | Gauss with $\sigma = 2$

- Extremely efficient to apply
- Easy to implement
- Blurs edges
- Not context aware
- Sensitive to outliers

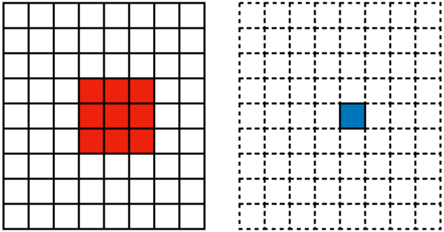
Remark

It is not possible to remove noise and not blur edges at the same time with a linear filter. You need non-linear filtering for this.

Subsection 2

Median filter

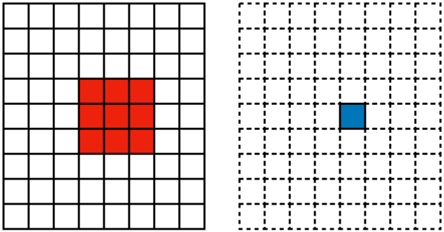
Median filter



- For each location, rank order all neighboring intensities
- Assign the median value as the result

$$1, 2, 2, 3, 2, 2, 9, 3$$

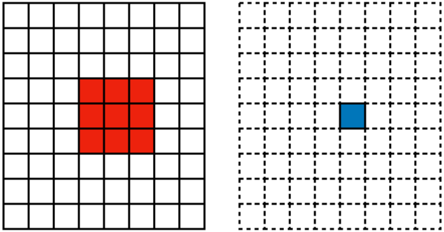
Median filter



- For each location, rank order all neighboring intensities
- Assign the median value as the result

$$1, 2, 2, 3, 2, 2, 9, 3$$
$$1, 2, 2, 2, 2, 3, 3, 9 \rightarrow 2$$

Median filter

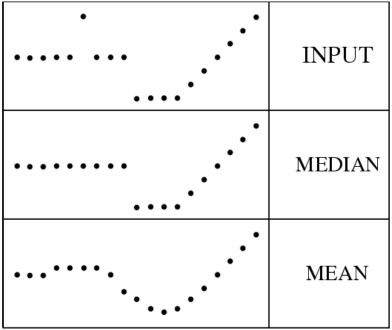


- For each location, rank order all neighboring intensities
- Assign the median value as the result

$$1, 2, 2, 3, 2, 2, 9, 3$$
$$1, 2, 2, 2, 2, 3, 3, 9 \rightarrow 2$$

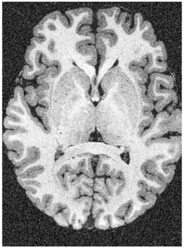
- Sliding window similar to convolution
- Other rank order filters, such as min and max, are defined similarly
- Non-linear filters
- Two rank filters are not commutative: $\min(\mathrm{median}(I)) \neq \mathrm{median}(\min(I))$

Properties



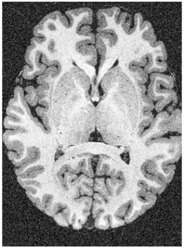
- Robustness to outliers - does not respond to spikes
- Preserves discontinuities, such as edges

Median filtering examples

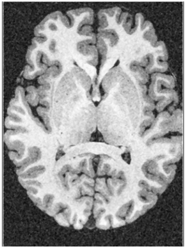


Noisy

Median filtering examples

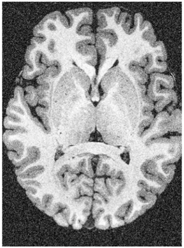


Noisy

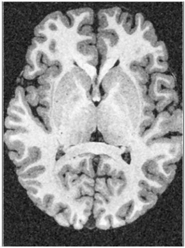


S_3

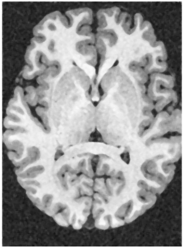
Median filtering examples



Noisy

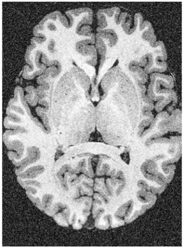


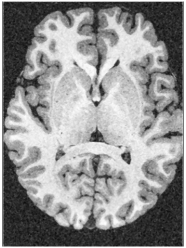
S_3

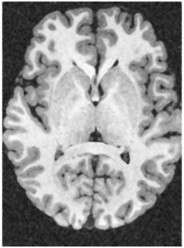


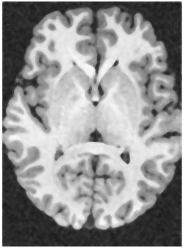
S_5

Median filtering examples

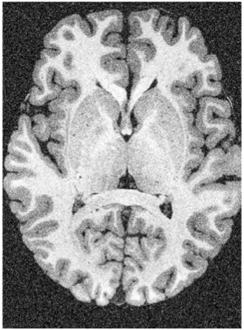

Noisy


S_3

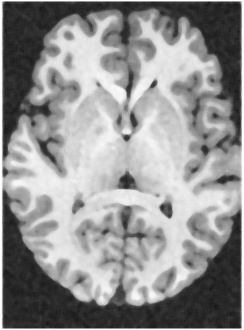

S_5


S_7

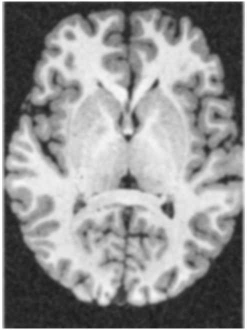
Analysis



Noisy



Median S_7



Gaussian S_7

- Robustness and edge preservation
- Easy and efficient implementation
- Patchy result
- Image content can change - adding and removing structures

Subsection 3

Non-local means

Non-local means

- [Buades, Coll and Morel, CVPR 2005]
- Very widely used model, very competitive results
- Extension of *bilateral filtering*

Non-local means

- [Buades, Coll and Morel, CVPR 2005]
- Very widely used model, very competitive results
- Extension of *bilateral filtering*
- Most noise suppression methods can be put in the form:

$$I(x) = \sum_{y \in \Omega} w(x,y) J(y) \text{ with } 0 \leq w(x,y) \leq 1 \text{ and } \sum_{y \in \Omega} w(x,y) = 1, \ \forall x$$

e.g. mean and Gaussian filter. *Weights* $w(x,y)$ changes accordingly.

Non-local means

- [Buades, Coll and Morel, CVPR 2005]
- Very widely used model, very competitive results
- Extension of *bilateral filtering*
- Most noise suppression methods can be put in the form:

$$I(x) = \sum_{y \in \Omega} w(x, y) J(y) \text{ with } 0 \leq w(x, y) \leq 1 \text{ and } \sum_{y \in \Omega} w(x, y) = 1, \ \forall x$$

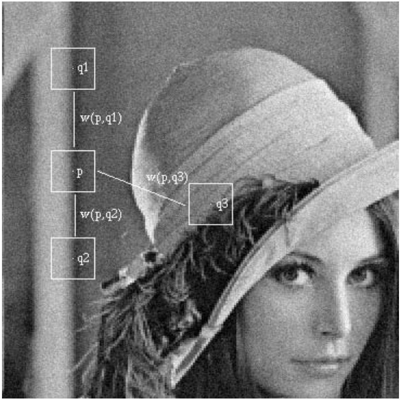
  e.g. mean and Gaussian filter. *Weights* $w(x, y)$ changes accordingly.

- Non-local means uses $w(x, y)$ to assess similarity between image information at x and y

$$w(x, y) = \frac{1}{Z(x)} \exp\left\{ -\frac{\| J(\mathcal{N}(x)) - J(\mathcal{N}(y)) \|_2^2}{h^2} \right\}$$

  $Z(x)$ is the normalization constant, h is the degree of filtering and $\mathcal{N}(x)$ neighborhood around x.

Non-local means - intuition



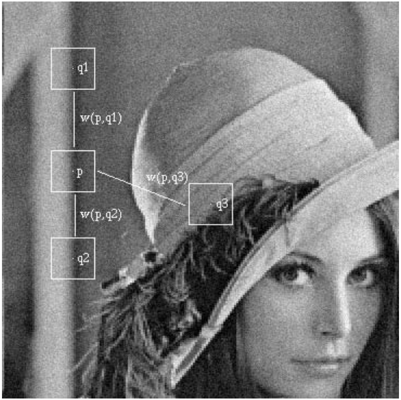
Taken from [Buades et al. CVPR 2005]

$$I(x) = \sum_{y \in \Omega} w(x,y) J(y)$$

$$w(x,y) = \frac{1}{Z(x)} \exp\left\{ -\frac{\|J(\mathcal{N}(x)) - J(\mathcal{N}(y))\|_2^2}{h^2} \right\}$$

- While smoothing point x use areas with similar image content
- If noise is random then averaging similar areas should get rid of it

Non-local means - intuition



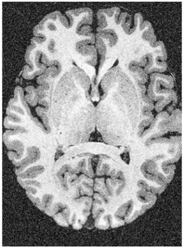
Taken from [Buades et al. CVPR 2005]

$$I(x) = \sum_{y \in \Omega} w(x,y) J(y)$$

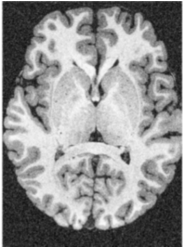
$$w(x,y) = \frac{1}{Z(x)} \exp\left\{ -\frac{\|J(\mathcal{N}(x)) - J(\mathcal{N}(y))\|_2^2}{h^2} \right\}$$

- While smoothing point x use areas with similar image content
- If noise is random then averaging similar areas should get rid of it
- Uses *redundancy* in the images, i.e. local structures repeat

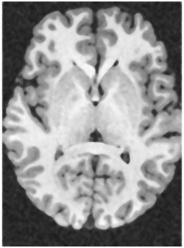
Results



Noisy

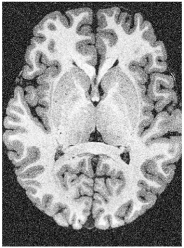


Binom with S_7

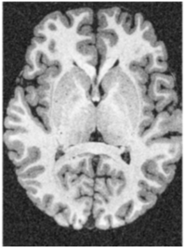


Median with S_7

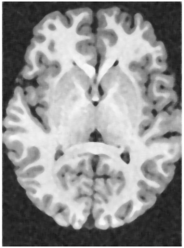
Results



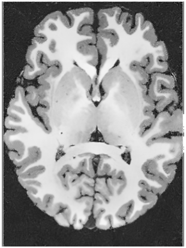
Noisy



Binom with S_7



Median with S_7



NLM $h = 15$

Properties

Noisy

- Increasing h leads to more smoothing

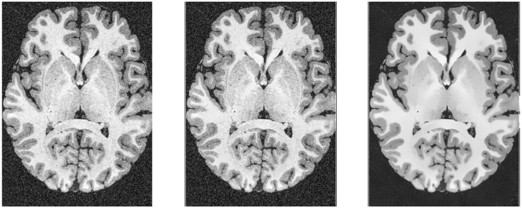
Properties

Noisy

$h = 5$

- Increasing h leads to more smoothing

Properties



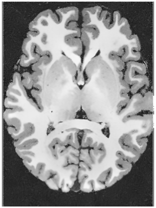
Noisy $\quad h = 5 \quad h = 15$

- Increasing h leads to more smoothing

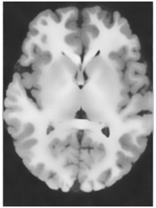
Properties

Noisy

$h = 5$



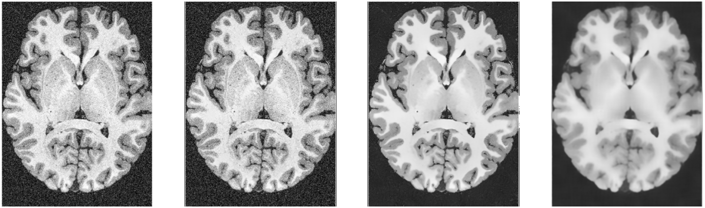
$h = 15$



$h = 50$

- Increasing h leads to more smoothing

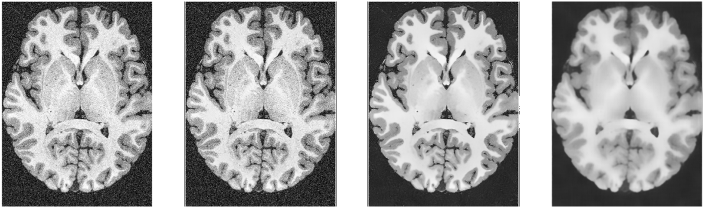
Properties



Noisy $h = 5$ $h = 15$ $h = 50$

- Increasing h leads to more smoothing
- Model parameters
  - h - depends on the noise level
  - size of $\mathcal{N}(x)$
  - size of $W(x)$

Properties

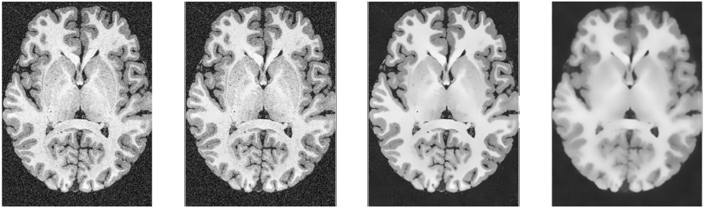


Noisy $h = 5$ $h = 15$ $h = 50$

- Increasing h leads to more smoothing
- Model parameters
  - h - depends on the noise level
  - size of $\mathcal{N}(x)$
  - size of $W(x)$
- The averaging often happens in a small *search-window*

$$I(x) = \sum_{y \in W(x)} w(x,y) J(y)$$

Properties



Noisy | $h = 5$ | $h = 15$ | $h = 50$

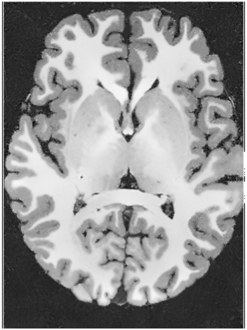
- Increasing h leads to more smoothing
- Model parameters
  - h - depends on the noise level
  - size of $\mathcal{N}(x)$
  - size of $W(x)$
- The averaging often happens in a small *search-window*

$$I(x) = \sum_{y \in W(x)} w(x, y) J(y)$$

- Size of $\mathcal{N}(x)$ matters:
  - $\mathcal{N}(x) = \Omega$ leads to mean filtering
  - $\mathcal{N}(x) = \{x\}$ leads to Yaroslavsky's filter, a special case of bilateral filtering

Analysis

- Higher noise suppression
- Better preservation of edges
- May create artifacts
- Important to estimate noise level - different algorithms to do it
  [Coupe et al. Medical Image Analysis 2010]
  ...
- Extension to MRI and diffusion MRI
  [Manjon et al. Medical Image Analysis 2008]
  [Wiest-Daessle et al. MICCAI 2008]
  ...
- Currently the state-of-the-art filtering technique that is not based on machine learning. We will see more about this later in the course.



Subsection 4

Energy-based

Optimization-based methods

Remember the additive model for the noisy image.

$$\underbrace{J(x)}_{\text{Observed}} = \underbrace{I(x)}_{\text{Desired}} + \underbrace{\epsilon(x)}_{\text{Noise}}$$

Optimization-based methods

Remember the additive model for the noisy image.

$$\underbrace{J(x)}_{\text{Observed}} = \underbrace{I(x)}_{\text{Desired}} + \underbrace{\epsilon(x)}_{\text{Noise}}$$

Denoising as an energy minimization problem

$$\hat{I} = \arg_I \min \lambda D(I, J) + R(I)$$

- $D(I, J)$: Data consistency
- $R(I)$: Regularization - embedding prior information (opinions) on the desired image.
- λ: Weighting coefficient.

Optimization-based methods

Remember the additive model for the noisy image.

$$\underbrace{J(x)}_{\text{Observed}} = \underbrace{I(x)}_{\text{Desired}} + \underbrace{\epsilon(x)}_{\text{Noise}}$$

Denoising as an energy minimization problem

$$\hat{I} = \arg_I \min \lambda D(I, J) + R(I)$$

- $D(I, J)$: Data consistency
- $R(I)$: Regularization - embedding prior information (opinions) on the desired image.
- λ: Weighting coefficient.

Alternative constrained optimization

$$\hat{I} = \arg_I \min R(I), \text{ such that } D(I, J) = 0$$

Total variation denoising

[Rudin, Osher and Fatemi, Physica D, 1992]

Remember the additive model for the noisy image.

$$\underbrace{J(x)}_{\text{Observed}} = \underbrace{I(x)}_{\text{Desired}} + \underbrace{\epsilon(x)}_{\text{Noise}}$$

Denoising as an energy minimization problem

$$\hat{I} = \arg_I \min \lambda D(I, J) + R(I)$$

- Data consistency:

$$D(I, J) = \int (I(x) - J(x))^2 \, dx = \|I - J\|_2^2$$

- Regularization:

$$R(I) = \int \|\nabla I\| dx = \|I\|_{TV}$$

- λ: Allowed noise level in the observed image with $\lambda = 1/2\sigma^2$

Total variation denoising: Experiments

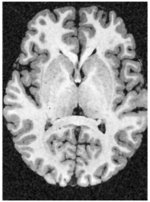
Total variation aims to get an image with minimal variation.

noisy image

Total variation denoising: Experiments

Total variation aims to get an image with minimal variation.

noisy image

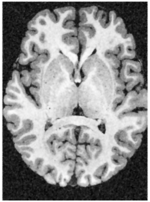


$\lambda = 0.2$

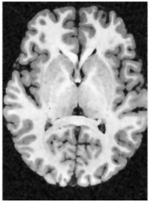
Total variation denoising: Experiments

Total variation aims to get an image with minimal variation.

noisy image



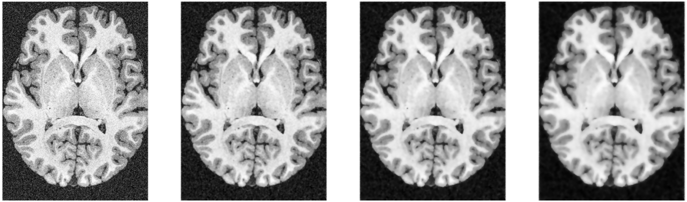
$\lambda = 0.2$



$\lambda = 0.3$

Total variation denoising: Experiments

Total variation aims to get an image with minimal variation.

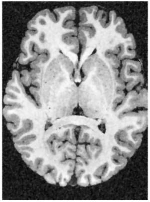


noisy image $\lambda = 0.2$ $\lambda = 0.3$ $\lambda = 0.5$

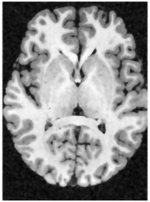
Total variation denoising: Experiments

Total variation aims to get an image with minimal variation.

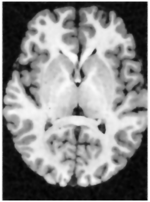
noisy image



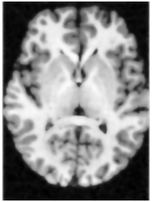
$\lambda = 0.2$



$\lambda = 0.3$



$\lambda = 0.5$



$\lambda = 1.0$

Probabilistic interpretation

The energy-based formulation can be interpreted in a probabilistic model. Remember that

$$\arg_I \min \lambda D(I, J) + R(I) = \arg_I \max \log P(J|I) + \log P(I)$$

- $P(J|I)$: Likelihood
- $P(I)$: Prior

Probabilistic interpretation

The energy-based formulation can be interpreted in a probabilistic model. Remember that

$$\arg_I \min \lambda D(I, J) + R(I) = \arg_I \max \log P(J|I) + \log P(I)$$

- $P(J|I)$: Likelihood
- $P(I)$: Prior

Total variation denoising model:

$$\begin{aligned} P(J|I) &= \mathcal{N}\left(J; I, \frac{1}{2\lambda}\right) \\ P(I) &= \frac{1}{C}\exp(-\|\nabla I\|) \end{aligned}$$

Remark

Note that the standard deviation of the likelihood model is linked to the weight in the energy-based formulation.

Generality

Note that the energy-based or the corresponding probabilistic formulations are very generic and many different models can be cast in the same formulation.

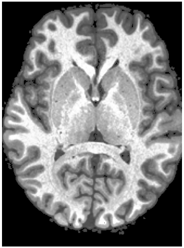
$$\arg_I \min \lambda D(I, J) + R(I) = \arg_I \max \log P(J|I) + \log P(I)$$

- $D(I, J)$ or $P(J|I)$ encode information how the noise process works. It does not have to additive Gaussian noise but it can be different things as we have seen before.
- $R(I)$ or $P(I)$ encode our prior information on what type of images are plausible.
  - TV norm: minimal variation
  - Wavelet: sparse in a transform domain
  - Dictionary learning: composed of sparse atomic elements.
  - Unsupervised learning: probabilistic models learned from data.
  - Adversarial learning: comparing distribution wise.

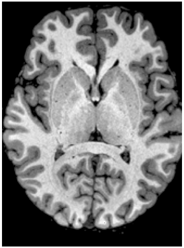
Section 5

Summary and Exercise

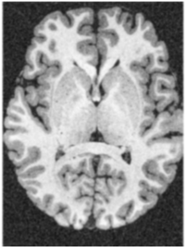
Summary



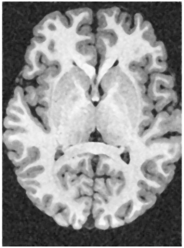
Histogram equalization



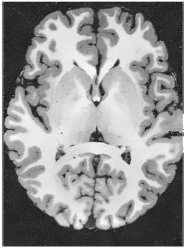
Intensity normalization



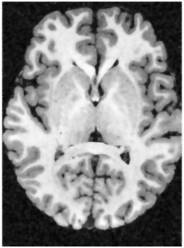
Linear filtering



Median filtering

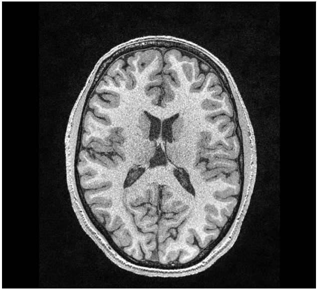


Non-local means

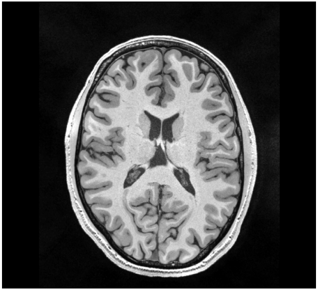


TV with $\lambda = 0.3$

Noise suppression challenge



Single acquisition



Average of 8

Challenge

High-resolution MRI is hard to acquire. One way is to acquire many images of the same subject and average them. 8 or so images yield good SNR but this leads to long acquisitions. Can you reduce this time with noise suppression?
More details in the moodle platform.

Outline - Next week

- Contrast enhancement
- Intensity normalization
- Noise suppression
- Exercise / Challenge
- Bias correction
- Variations in image and pixel-size
- Discussions about the exercise