Light Microscopy in 4 Dimensions: Automatic Image Processing and Segmentation

Jussi Tohka
jussi.tohka@tut.fi

Institute of Signal Processing
Tampere University of Technology

About the lectures

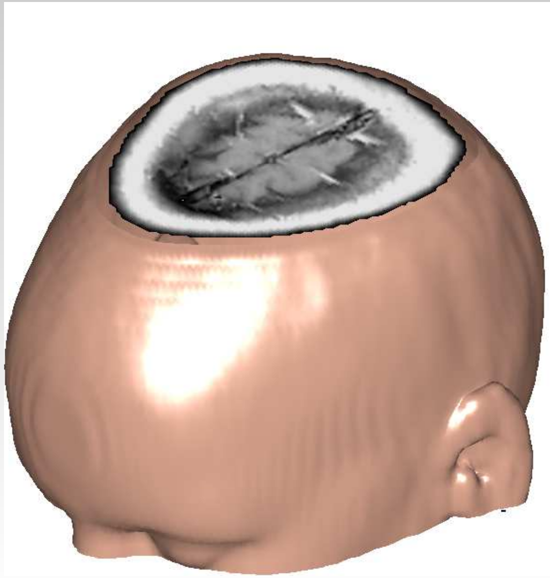
Goal: Give an overview of basic techniques used in automatic processing and segmentation of 3-D biomedical images

- Only general principles are presented, references are offered for technical details.
- No particular problem domain (e.g. light microscopy) is considered, techniques presented will be quite general.
- Most techniques presented can be applied to process also plane images, but the presentation and symbolics will refer to three dimensions.

Overview

- **Basics**: Some 3D imaging modalities. The definition of image, differences between natural and biological images, differences between plane and 3-D images.
- **Basic image processing**: Image filtering and smoothing in three dimensions. Edge detection in three dimensions.
- **Segmentation**: The definition of the image segmentation. Image models and pattern classification.
- **Typical imaging artifacts**: Intensity non-uniformity. Partial volume effect and blurring.
- **Spatial information for image segmentation**: Markov Random Fields, Deformable models, Gradient flows.

3D images



3D imaging modalities

- **Microscopy**
- **MRI**: (Anatomical) magnetic resonance imaging provides accurate 3-D representations of the anatomy. Medical and biological applications numerous. Functional MRI is useful for studying activations (in brain). Also, e.g. Diffusion (tensor) MRI and MR angiography.
- **CT**: Computerized tomography provides anatomical information, good contrast between bones and soft tissues.
- **PET and SPECT**: Physiological information. Besides clinical applications, useful for drug development, brain research etc.
- **3D ultrasound**

3D imaging modalities

`laxmi.nuc.ucla.edu:8248`
`/M248_98/intro/medimage.html`

Image related operations

- **Image processing**: image in $\longrightarrow$ image out
- **Image analysis**: image in $\longrightarrow$ numbers out
- **Image understanding**: image in $\longrightarrow$ high level description out
- **Graphics**: (3-D) Image(s) in $\longrightarrow$ 3-D visualization, movie etc. out

During this course the goal is **image analysis**, but to get there we need **image processing**.

Image analysis paradigm

Pre-processing is needed to ease the segmentation task. **Segmentation** means partitioning image domain into different subsets. It is needed because rarely the whole image represents the thing we are interested about. Automatic image segmentation methods are the focus during these lectures.

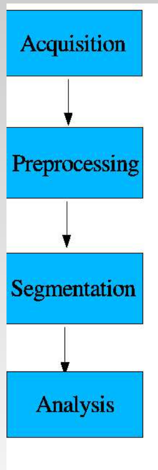
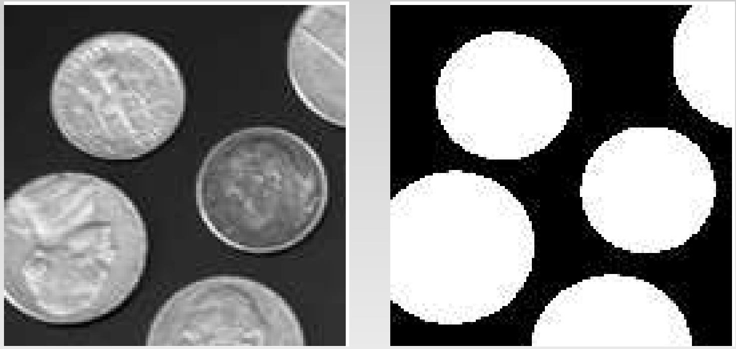


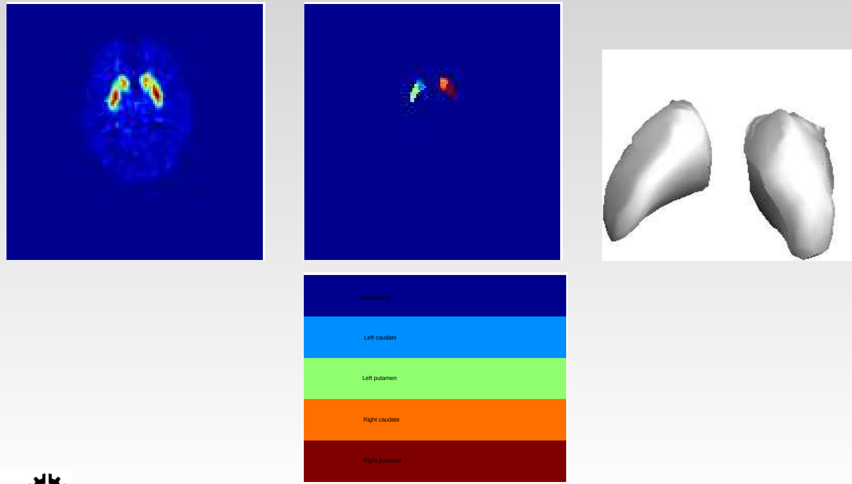
Image segmentation examples



- Black = background
- White = Coin

Image segmentation examples

This example is drawn from the automatic analysis of Raclopride-PET images. Ref: Tohka *et al*: Improved Reproducibility in D2-Receptor Studies with Automatic Segmentation of Striatum from PET Images, to appear in IEEE Medical Imaging Conference (MIC2004), 2004.



Quantitative image analysis

- **Image analysis** referred to computing numerical values based on images
- **Quantitative** means these values have some actual biological meaning, and that they can be compared across the images.
- The importance of the **quantitative** image analysis that it allows comparisons between 1) two different images, 2) two groups of images, 3) a single image and a group of images, 4) a single image to the expectation (diagnosis).
- Some examples of the applications of quantitative analysis within microscopy:
  - Diagnosis of the cervical cancer (B. L. Luck *et al* IEEE-ICIP 2003)
  - Pollen rate measurement (P. Bonton *et al* Image Anal Stereol. 2001)

Why to automate segmentation?

- The sheer size of images makes the manual segmentation time consuming and not cost-effective; even for professional with good equipment it usually takes several hours to perform the required segmentation task.
- The reproducibility of manual segmentation is poor even in the intra-observer case. See Zijdenbos, Forghani, Evans: Automatic quantifi cation of MS lesions in 3D MRI brain data sets: Validation of INSECT, In proc. of MICCAI98.
- There are tasks involving calculations for which computers do much better than humans.

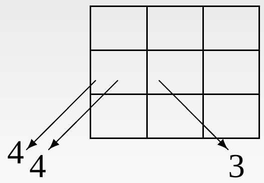
Images

An image is a measured physical quantity, here referred as the intensity, as a function of position. Image intensities can be almost anything, and **need not to be scalar valued**. Images can represented as

- arrays of intensity values, or
- functions from the image domain to the set of intensity values

Here we use both representations interchangeably. A figure explains...

1	2	3
4	3	2
1	5	0

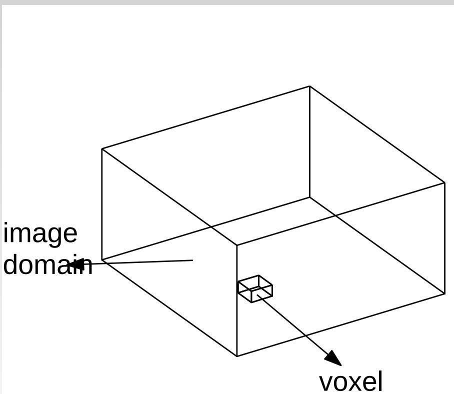


Volumetric images

- Volumetric images are 3-D arrays, or functions from $\mathbb{R}^3$ to the set of intensity values.
- The image domain is sub-divided into elementary parts termed *voxels*.
- The intensity at the voxel i, j, k, $\mathcal{V}_{ijk}$ is given by either I_{ijk} or $I(\mathbf{x})$ with $\mathbf{x} \in \mathcal{V}_{ijk}$.

Volumetric images

- Intensity for a voxel i, j, k, $\mathcal{V}_{ijk}$ is given by either I_{ijk} or $I(\mathbf{x})$ with $\mathbf{x} \in \mathcal{V}_{ijk}$



2-D or 3-D. How do they differ?

- In principle, 2-D (plane) images and 3-D (volumetric) images are not any different except for additional co-ordinate in the image domain (peanuts!).
- However,
  1. 3-D images contain typically more data $\longrightarrow$ image analysis becomes computationally more burdensome.
  2. Topology, and hence geometry, in 3-D is more 'rich' than in 2-D. The image topology here means structural properties of the image, for example statements like cerebral cortex is within the brain volume are topological in nature. $\longrightarrow$ If we know something about the image before seeing it, in the volumetric case there usually are some extra constraints that can be set.

Biological images vs. natural images

- With natural images, the ultimate task of image analysis is often **recognition** that means deciding what is in the scene. For example, if somebody sees a horse or a lion, he certainly acts differently. He is not interested whether the lion is big or not.
- With biological images, it is known what there is in the images. For example, if we have an MR image of an human head , so there should be a human head in it, which hopefully contains also a brain.
- It is of interest how this head looks like. For example, interest could be how many of something there is, what shape does a particular structure have.
- **The questions are more specific!**

Dude, where's my 4th dimension?

- Sometimes a study consist of series of (volumetric) images acquired at different times, or during different frames of time. These images are here said to be dynamic.
- Dynamic images can describe
  - 1) the movement of something within the image domain,
  - or 2) a change in intensity values with time.
- Case 1): The time dimension adds to dimensionality of the image domain, thus volumetric dynamic images are 4-D images. Time can be considered as similar dimension as others (so-called spatial dimensions) but it can also have some specific features.

Dude, where's my 4th dimension?

- Case 2: The image domain is not larger than with static images and intensity values at each voxel are now vectors instead of scalars. That is, we have more information for segmentation task. In this case, movement within image is a major problem.

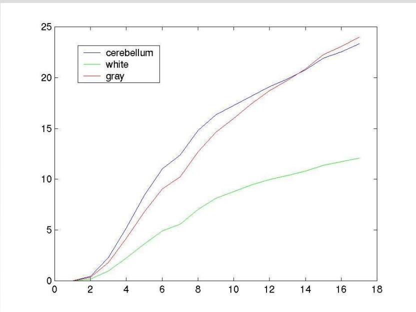


IMAGE PRE-PROCESSING

Gaussian filtering in 3 dimensions

Filtered image $\hat{I}$

$$\hat{I} = I * G, \hat{I}(x, y, z) = \int\int\int I(x-r, y-s, z-t)G(r, s, t)drdsdt,$$

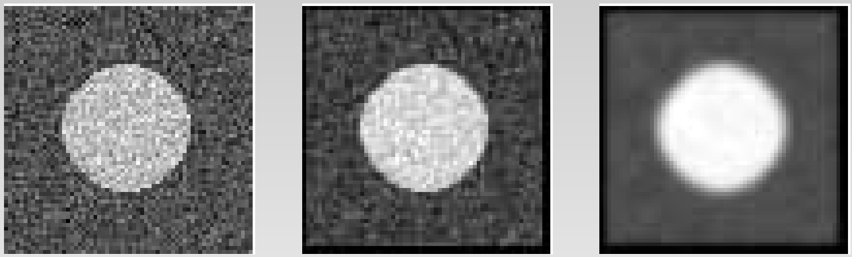
where $G : \mathbb{R}^3 \to \mathbb{R}$ is the Gaussian kernel.

- Pros: Simple, fast, effective in the suppression of the white Gaussian noise
- Cons: Does not preserve edges

Discrete Matlab implementation

```
function filimg = gaussian3Dfil(img,siz,std);
    % construct the kernel;
    [x,y,z] = meshgrid(-(siz(2)-1)/2:(siz(2)-1)/2...
      ,-(siz(1)-1)/2:(siz(1)-1)/2,-(siz(3)-1)/2:(siz(3)-1)/2);
    ker = exp(-(x.*x + y.*y + z.*z)/(2*std*std));
    ker = ker/sum(sum(sum(ker)));
    %filter
    filimg = convn(img,ker,'same');
```


Iterated Gaussian filtering



Original | one iteration | 10 iterations

Heat equation

- Iterated Gaussian filtering may be regarded as a solution to the heat-equation (or diffusion equation)

$$\frac{\partial I(\mathbf{x};t)}{\partial t} = \mathrm{div}\nabla I(\mathbf{x};t)) = \nabla^2 I(\mathbf{x},t) = (I * G)(\mathbf{x},t).$$

$$I(\mathbf{x},0) = I(\mathbf{x}).$$

- This is basis of the (linear) scale space, and also to some more advanced filtering schemes.

3-D anisotropic diffusion

A filtering and edge-detection scheme that preserves the edges and is effective in reducing noise in homogeneous image volumes. Primary references:

1. Perona, Malik: Scale space and edge detection using anisotropic diffusion, IEEE-TPAMI 12:629 - 639, 1990.
2. Gerig *et al.*: Nonlinear anisotropic filtering of MRI data, IEEE-TMI 11:221 - 232, 1992.

3-D anisotropic diffusion

The smoothing is formulated as a diffusive process. The filtered image is the image $I(\mathbf{x}, \infty)$ when the image 'evolves' by (in practise, the iteration has to stopped at some point)

$$\frac{\partial I(\mathbf{x};t)}{\partial t} = \mathrm{d}iv(c(\mathbf{x};t)\nabla I(\mathbf{x};t)); I(\mathbf{x},0) = I(\mathbf{x})$$

and

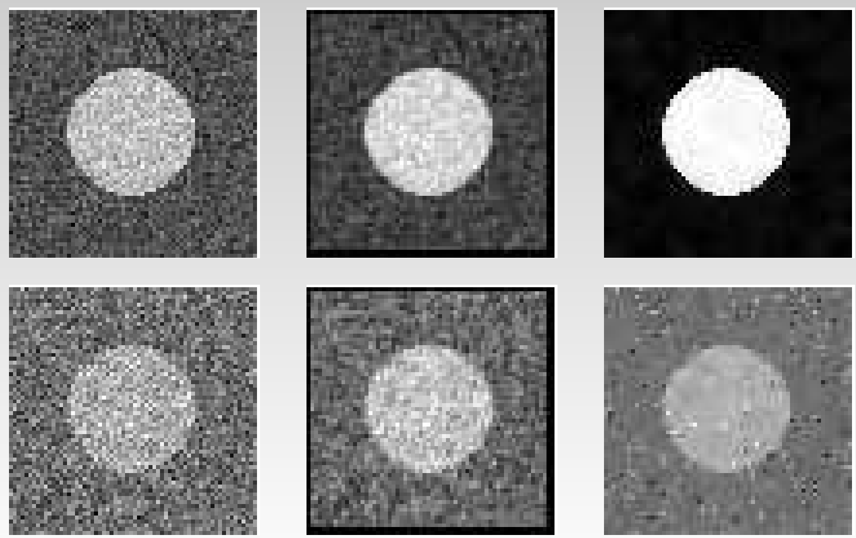
$$c(\mathbf{x};t) = \exp(-(\nabla I(\mathbf{x};t)/K)^2);$$

$$\mathrm{d}iv I = \frac{\partial I}{\partial x} + \frac{\partial I}{\partial y} + \frac{\partial I}{\partial z}.$$

A discrete Matlab implementation

```
function filimg = difffilter3D(img,K,N)
  delta_t = 1/7.1; [Nx,Ny,Nz] = size(img);
  for t = 1:N
    Ixp=cat(1,img(1,:,:), img(1:Nx-1,:,:))-img;
    Ixn=cat(1,img(2:Nx,:,:),img(Nx,:,:))-img;
    Iyp=cat(2,img(:,2:Ny,:), img(:,Ny,:))-img;
    Iyn=cat(2,img(:,1,:), img(:,1:Ny-1,:))-img;
    Izp=cat(3,img(:,:,1), img(:,:,1:Nz-1))-img;
    Izn=cat(3,img(:,:,2:Nz),img(:,:,Nz))-img;
    cxp = exp(-(abs(Ixp)/K).^2);
    cxn = exp(-(abs(Ixn)/K).^2);
    cyp = exp(-(abs(Iyp)/K).^2);
    cyn = exp(-(abs(Iyn)/K).^2);
    czp = exp(-(abs(Izp)/K).^2);
    czn = exp(-(abs(Izn)/K).^2);
    img = img + delta_t*(Ixp.*cxp + Ixn.*cxn + Iyp.*cyp + ...
                         Iyn.*cyn + Izp.*czp + Izn.*czn);
  end
  filimg = img;
```

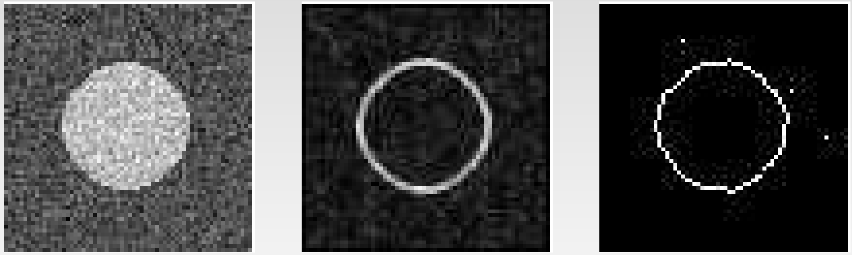

Gaussian vs. AIF



original Gaussian AIF

3D edge detection

- The aim: To locate places where image intensity level changes significantly.
- Edge detection can produce either a binary image, or an intensity image which only quantifies how strong is the change in the intensity value:



The 3-D Sobel edge operator

- Replica of the 2-D Sobel operator, optimal in some circumstances
- Results in images of edges in x, y, and z directions. Edge image in x-direction is obtained as

$$\nabla_x I(x, y, z) = \int\int\int \phi_x(r, s, t) I(x - r, y - s, z - t) dsdrdt,$$

$$\phi_x(r, s, t) = r/\sqrt{r^2 + s^2 + t^2}.$$

- The reference: Zucker and Hummel: A three-dimensional edge operator, IEEE-TPAMI, 1981.

Matlab implementation

```
function sobelimg = sobel3D(img)
  sz = size(img);
  kernelx = zeros(3,3,3);
  kernelx(3,1:3,1:3) = [sqrt(3)/3 sqrt(2)/2 sqrt(3)/3
                        sqrt(2)/2 1 sqrt(2)/2
                        sqrt(3)/3 sqrt(2)/2 sqrt(3)/3];
  kernelx(1,1:3,1:3) = -kernelx(3,1:3,1:3);
  kernely = permute(kernelx,[2 1 3]);
  kernelz = permute(kernelx,[3 2 1]);
  sobelx = convn(img,kernelx,'valid');
  sobely = convn(img,kernely,'valid');
  sobelz = convn(img,kernelz,'valid');
  sobelimg = zeros(sz);
  sobelimg(2:sz(1) - 1,2:sz(2) - 1,2:sz(3) -1) = ...
      sqrt(sobelx.*sobelx + sobely.*sobely + sobelz.*sobelz);
```

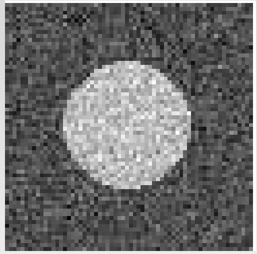

IMAGE SEGMENTATION

Notation

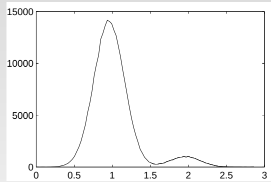
- Because indexing voxels with three indices is a pain, I will often use only a single index
- The image domain $\mathcal{D} = \{\mathcal{V}_1, \ldots, \mathcal{V}_K\}$. The set of voxel indices $\{1, \ldots, K\}$ is denoted by S. Each voxel $\mathcal{V}_i$ has an intensity value a_i. The image intensities (the data) is $\mathbf{a} = [a_1, \ldots, a_K]$. The random variable (RV) relating to the intensity of the voxel i is A_i.
- When segmenting images, we assign a label for each voxel based on the intensity values. The label says that this voxel is part of some structure or background. The set of labels includes whatever we want to extract from images and background.
- Integers $1, \ldots, L$ denote the labels. The segmented image is then $\mathbf{b} = [b_1, \ldots, b_K]$, where b_i - the label of the voxel $\mathcal{V}_i$ - is some integer from 1 to L.

Segmentation: thresholding

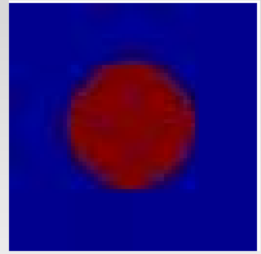
The simplest approach to the image segmentation is thresholding. It usually is not very useful in itself but it is the foundation for many more advanced segmentation methods.



image



histogram



segmentation

Histogram: The grouping of image intensities into bins (spaced apart by the so-called class interval) plotting the number of members in each bin versus the bin number. Can be viewed as an estimate of the probability density of the image intensity.

Thresholding

- How to select the thresholds correctly and optimally?
- It is assumed that we know the number of regions image is segmentated to. Let this be L.
- Then selecting the threshold values becomes simply a problem of pattern classification: The optimal selection minimizes the probability of error.

Image models

- To be able to select the thresholds optimally, we need a statistical model of the image intensities.
- Physics behind the image acquisition can be used to build the statistical model.
- For example, we can assume that the intensity values for each structure are normally distributed.
- This leads to **Gaussian mixture model** (GMM):

$$p(a) = \sum_{l=1}^{L} p_l g(a; \mu_l, \sigma_l) = \sum_{l=1}^{L} p_l \frac{1}{\sqrt{2}\sigma_l} \exp(-\frac{(a-\mu_l)^2}{2\sigma_l^2});$$

$$\sum_{l=1}^{L} p_l = 1.$$

Selecting optimal thresholds

- Under the GMM assumption, the optimal segmentation is obtained by selecting

$$b_i = l \quad \text{if} \quad p_l g(a_i; \mu_l, \sigma_l) > p_k g(a_i; \mu_k, \sigma_k) \forall k \neq l.$$

- This is just a version of the Bayes decision rule that states : assign the label l to an object with the feature vector $\mathbf{f}$ if the probability of l given $\mathbf{f}$ is greater than for any other label.

Parameter estimation

- There are parameters in the GMM whose values are not known us:
  1. μ_l; means of the intensity values for each structure.
  2. σ_l; variances of the intensity values for each structure.
  3. p_l; prior probabilities, which model the percentage of voxels belonging to the structure l.
- These can sometimes be solved based on exemplars from the manual segmentations (supervised learning). But, in some imaging modalities variations within individuals (and machinery) are so strong that supervised learning approaches are useless.

Unsupervised learning

- Learn μ_l, σ_l, p_l from the data itself.
- Various approaches: expectation maximization algorithm for maximum likelihood parameter estimates is among the most general ones.
- Reference: Dempster et al: Maximum likelihood from incomplete data via the EM algorithm, J Royal Stat Soc 39:1 - 38, 1977
- If only the means are unknown and variances as well as prior probabilities are equal for each structure, then K-means algorithm is the choice.

K-means clustering

- A simple method for clustering is K-means clustering, which tries to minimize

$$J = \sum_{j=1}^{L} \sum_{b_i = j} ||a_i - \mu_j||^2$$

  to find the cluster centers c_j.
- The algorithm: Repeat
  1. Set $\mu_j = \sum_{b_i = j} a_i$ for all j.
  2. Select $b_i = min_j ||a_i - \mu_j||^2$ for all i.

Thresholding may not work..

- because the image model is too simple. This is usually the case with GMMs because
  1. intensity non-uniformity
  2. partial volume effect and image blur
- because the GMM image model does not account for spatial geometry of the image. That is, in GMM, each voxel value is assumed to be independent, and their spatial dependence is not taken into account.

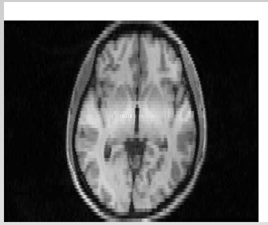
Intensity non-uniformity

- Exists at least in MRI (due to complex electromagnetic phenomena) and in the light microscopy (due to varying level of the illumination across the field-of-view).
- The retrospective correction procedures usually assume that non-uniformity is additive or multiplicative Gaussian noise with a strong spatial correlation.
- Image model then becomes:

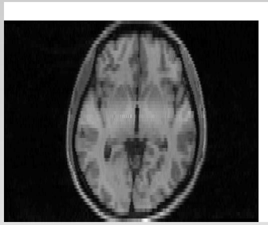
$$A_i = U_i + N_i \quad \text{or} \quad A_i = U_i N_i,$$

  where A_i is RV of the intensity of $\mathcal{V}_i$, U_i is the intensity without non-uniformity, and N_i is the intensity non-uniformity at $\mathcal{V}_i$.
- U_i can be modeled by a GMM.

Example



original MR image



non-uniformity corrected

- Reference: Sled, Zijdenbos, Evans: A Nonparametric method for automatic correction of intensity nonuniformity in MRI data, IEEE-TMI 17:87 -97

Partial volume effect

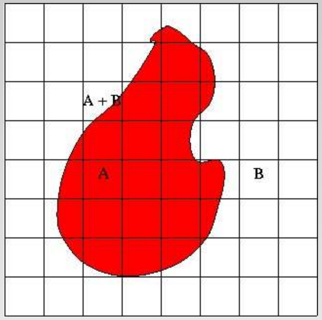
- The intensity value within a voxel may originate from two or more structures.
- This can be due to the image blurring or simply due to finite imaging resolution.
- In both cases, image intensities can be modeled as

$$U_i = \sum_{l=1}^{L} V_{il} T_l;$$

$$A_i = \sum_{l=1}^{L} V_{il} T_l + N_i$$

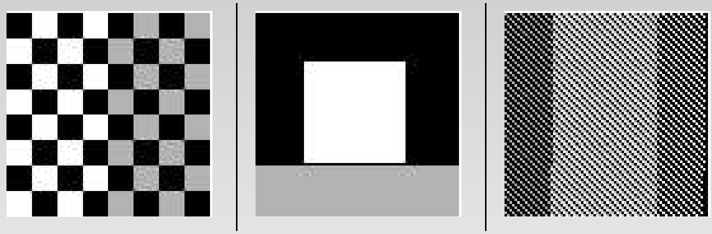
  Note that this is **not** a GMM!

Partial volume effect



Histogram limitations

All the images above have the same histograms.



This is meant to say that image histogram does not include much information about the content of image, and therefore thresholding is of limited use when segmentating the images.

Histogram limitations