

Chapter 11

How can physicians quantify brain degeneration?

*“If the human brain were so simple that we could understand it,
we would be so simple that we couldn't” E. M. Pugh*

M. Bach Cuadra([°]), J.-Ph. Thiran([°]), F. Marques(^{*})

([°]) Ecole Polytechnique Fédérale de Lausanne, Switzerland

(^{*}) Universitat Politècnica de Catalunya, Spain

Life expectation is increasing every year. Along with this aging population, the risk of neurological diseases (e.g. dementia¹) is considerably increasing² as well. Such disorders of the human nervous system affect the patient from both a physical and social point of view, as in the case of Alzheimer's disease, one of most known brain disorders (Mazziotta, Toga et al. 2000).

To pose their diagnosis, in addition to the usual clinical exam, physicians increasingly rely on the information obtained by means of 2D or 3D imaging systems: the so-called *image modalities*. Nowadays there exists a wide range of 3D medical image modalities that allow neuroscientists to see inside a living human brain. 3D imaging is of precious help since it allows, for instance, the physician to better localize specific areas inside the

¹ Dementia is a progressive degenerative brain syndrome which affects memory, thinking, behavior and emotion.

² In 2006, 24.3 million people in the world have dementia, with 4.6 million new cases annually. By 2040 the number will have risen to 81.1 million. Source: <http://www.alz.co.uk/>

body and to understand the relationships between them. Among all medical image modalities, Magnetic Resonance Imaging (MRI) has been increasingly used since their invention in 1973 by P. Lauterbur and Sir P. Mansfield³. As a matter of fact, MR imaging has useful properties: it provides an anatomical view of the tissues and deep structures of the body (see Fig. 11.1) thanks to the magnetic properties of the water, without the use of ionic radiation (Haacke, Brown et al. 1999).

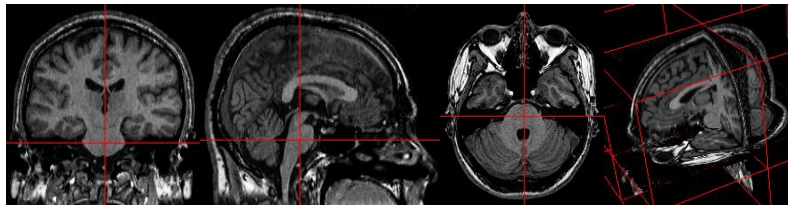


Fig. 11.1 3D MR image of a human brain: coronal, sagittal, axial and 3D views. The red (dark grey) cross indicates the same spatial point in the three views.

For a long time MRI has been used qualitatively, for visualization purpose only, to help physicians in their diagnosis or treatment planning. For instance when physicians look at MR images to evaluate lesions or tumors, they first segment the suspicious growth mentally and then use their training and experience to assess its properties. However, due to several factors such as the ever-increasing amount of clinical data generated in clinical environments and their increasingly complex nature, it is nowadays necessary to rely on *computerized segmentation techniques* (Suri, Wilson et al. 2005). By segmentation we mean dividing an image into homogeneous regions for a set of relevant properties such as color, intensity or texture which allow a quantitative analysis of medical images (automatic estimation of quantitative measures and regions of interests).

In the specific case of brain disorders, accurate and robust brain tissue segmentation from MR brain images is a key issue for quantitative studies (Bach Cuadra, Cammoun et al. 2005). Studying MR brain images of such patients is a key step, first to differentiate from a normal tissue loss due to normal aging; second, to better understand and characterize the anatomy of such patients; and third, to better assess drug treatments. Thus, there is a need in quantifying MR image changes to determine *where* the tissue loss is and its *amount*.

³ They were awarded the 2003 Nobel Prize in Medicine or Physiology for their discoveries concerning MRI.

In this Chapter we will address the problem of quantifying brain degeneration in a patient suffering from focal brain degeneration, causing a progressive aphasia. To do so, we will do a longitudinal study, that is, we will compare two MR images from the same patient acquired in a year interval (I1 in 2000 and I2 in 2001). First, the image classification problem must be solved for every 3D image acquisition (see Fig. 11.2). Second, to detect and quantify the tissue loss we need to compare the I1 segmentation with the I2 segmentation in a point to point basis. Therefore, an *image registration* problem arises: a point-to-point correspondence is not ensured between both MR scans since the patient has not been lying in the same position during the two acquisition process. This can be solved by a rigid registration process which looks for the spatial transformation that will bring one image to a point-to-point correspondence with the other one. In this application, only translation and rotation are needed to compensate for a different position of the patient in the two scans. Even in such a simplified framework (only translation and rotation), image registration is a very complex problem (Hajnal, Hill et al. 2001) whose complete discussion is out of the scope of this Chapter. For the sake of completeness, the registration problem is briefly addressed in Section 11.3.

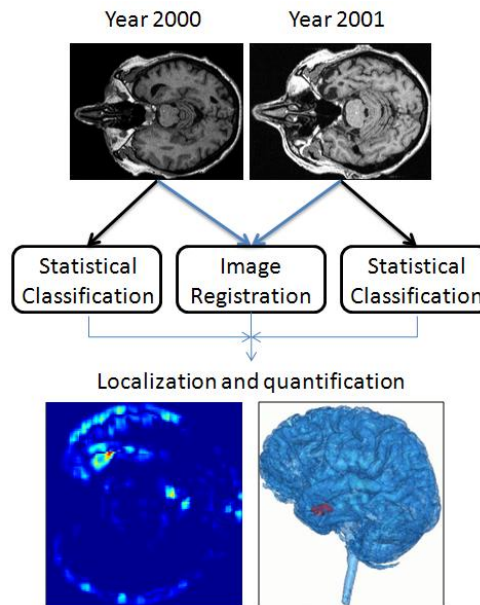


Fig. 11.2 Image processing tools presented in this Chapter

The Chapter is organized as follows. In Section 11.1 we present the theoretical framework to solve the image classification problem. In Section 11.2 a MATLAB proof of concept for the image classification will be developed. As previously commented, the problem of image registration is briefly discussed in Section 11.3. Finally, in Section 11.4 a discussion concludes the Chapter.

11.1 Background 1 – Statistical Pattern Recognition for Image Classification

Numerous approaches have been proposed for image classification. They can be divided into two main groups: (i) supervised classification, which explicitly needs user interaction, while (ii) unsupervised classification is completely automatic. In this Chapter we describe an unsupervised statistical classification approach. In fact, this can be seen as an estimation problem: estimate the underlying class (*hidden data*) of every voxel (3D extension of the pixel concept) from the 3D intensity image (*observed data*). This concept is illustrated in Fig. 11.3.

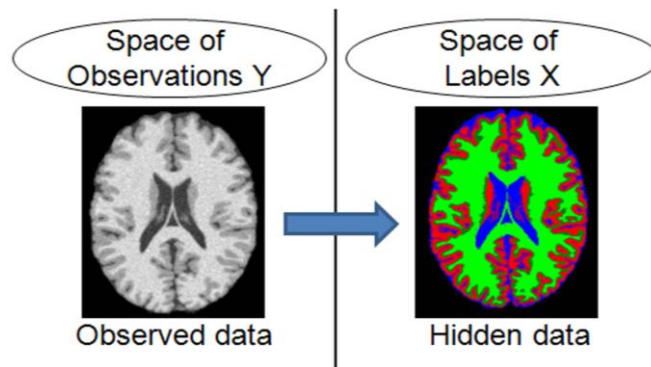


Fig. 11.3 Classification problem seen as an estimation problem. MR image intensities (on the left) are the observed data. The labeled image (on the right side) is what we are looking for: blue (dark grey) voxels are the Cerebrospinal Fluid class (CSF), red (medium grey) voxels are Grey Matter (GM), and green (light grey) voxels are White Matter (WM).

11.1.1 Statistical framework

Formally, let us consider N data points (typically voxels) and the observed data features $y_i \in \mathbb{R}$, with $i \in S = \{1, 2, \dots, N\}$ ⁴. In the MR image case, Y_i is the random variable associated to the intensity of voxel i , with the set of possible outcomes D and y_i represents the observed intensity of voxel i . The ensemble of all random variables defining the MR image random process is denoted by Y and any simultaneous observation of the random Y_i , (that is, any observed MR image) is denoted by

$$\mathbf{y} = \{y_1, y_2, \dots, y_N\} \subset D^N \subset \mathbb{R}^N \quad (11.1)$$

The classification process aims at classifying the data into one of the hidden underlying classes labeled by one of the symbols $L = \{CSF, GM, WM\}$. CSF stands for Cerebrospinal Fluid (blue or dark grey voxels in Fig. 11.3), GM for Grey Matter (red or medium grey voxels in Fig. 11.3), whereas WM for White Matter (green or light grey voxels in Fig. 11.3). X_i is the random variable associated to the class of voxel i , with the set of possible outcomes L , and x_i represents the class assigned to voxel i . The ensemble of all random variables defining the hidden underlying random process is denoted by X and any simultaneous configuration of the random variables X_i , (that is, any label map) is given by

$$\mathbf{x} = \{x_1, x_2, \dots, x_N\} \subset L^N \quad (11.2)$$

In this Chapter, we are looking for the *most probable* label map \mathbf{x}^* given the image observation \mathbf{y} . As already introduced in Chapter 4, this can be expressed mathematically by the so-called *Bayesian*, or *Maximum a Posteriori* (MAP), decision rule as⁵:

⁴ S represents the grid in which voxels are defined and, therefore, i is a specific site in this grid

⁵ As in Chapter 4, here we will often use a shortcut notation for probabilities, when this does not bring confusion. The probability $P(A=a|B=b)$ that a discrete random variable A is equal to a given the fact that random variable B is equal to b will simply be written $P(a/b)$. Moreover, we will use the same notation when A is a *continuous* random variable for referring to *probability density* $p_{A|B=b}(a)$.

$$\mathbf{x}^* = \arg \max_{\mathbf{x} \in X} P(\mathbf{x} | \mathbf{y}, \Theta) \quad (11.3)$$

where \mathbf{x} represents a *possible* label map and the conditional probability is evaluated over all possible label maps X , and Θ represents the set of parameters used to estimate the probability distribution. For instance, in the Gaussian case, these parameters are the mean vector and the covariance matrix of the process.

Bayes' rule can be then applied to Eq. (11.3), yielding:

$$P(\mathbf{x} | \mathbf{y}, \Theta) = \frac{P(\mathbf{y} | \mathbf{x}, \Theta)P(\mathbf{x})}{P(\mathbf{y} | \Theta)} \quad (11.4)$$

where $P(\mathbf{y} | \mathbf{x}, \Theta)$ represents the contribution of the so-called *intensity image model* (i.e., the likelihood that a specific model \mathbf{x} has produced the intensity image observation \mathbf{y}) that models the image intensity formation process for each tissue type; $P(\mathbf{x})$ represents the contribution of the so-called *spatial distribution model* (i.e., the a priori probability of the corresponding label map) that models the spatial coherence of the images, and $P(\mathbf{y} | \Theta)$ stands for the a priori probability of the observed image.

Based on the above, we thus have to address the following *classification* problem. Given an image intensity \mathbf{y} , find the most probable label map \mathbf{x}^* such that:

$$\mathbf{x}^* = \arg \max_{\mathbf{x} \in X} \frac{P(\mathbf{y} | \mathbf{x}, \Theta)P(\mathbf{x})}{P(\mathbf{y} | \Theta)} \quad (11.5)$$

The conditional probability $P(\mathbf{y} | \Theta)$ is often assumed to be constant for all hypotheses of label maps and, therefore, it can be ignored in the maximization process, so that Eq. (11.5) simplifies to:

$$\mathbf{x}^* = \arg \max_{\mathbf{x} \in X} P(\mathbf{y} | \mathbf{x}, \Theta)P(\mathbf{x}) \quad (11.6)$$

which itself implies two steps:

- *Image intensity modeling*: Image modeling refers to the estimation of $P(\mathbf{y} | \mathbf{x}, \Theta)$, and this is typically carried out using Gaussian Mixture Models (GMM; see Section 11.1.2). As we will see, assuming data independency, classification can be performed relying only on this model. However, this assumption is too strong and the resulting

classifiers lead to low quality classification results (see Section 11.2.3).

- *Spatial distribution modeling*: Spatial prior probabilities of classified images $P(\mathbf{x})$ are modeled using Markov Random Fields (MRF; see Section 11.1.4).

Actually, the global model to be used comes from the combination of the two previous models. This way, in Section 11.1.5, we will introduce the concept of Hidden Markov Random Fields (HMRF) that allows including dependency among random variables. Finally, in Section 11.1.6, Gaussian Hidden Markov Random Fields (GHMRF) will be presented as a specific case of the previous HMRF model.

Note that there are in fact many aspects of the theory that we will present here that are shared with Chapter 4. The same MAP method is used, thus the mathematical formalism is almost the same. As we shall see, however, some differences exist: first, here we will deal with 3D data; and second, the prior probability $P(\mathbf{x})$ will be modeled by Hidden Markov Random Fields where emission probabilities will not be estimated as in Chapter 4, but modeled by an exponential function thanks to the Hammond-Clifford theorem.

11.1.2 Gaussian Mixture Models (GMM)

We define in this Section the initial mathematical model of the intensity distribution of an MR image volume. Let us suppose, for now, that all the random variables representing the image intensity, Y_i (with $i \in S$), are identically and independently distributed. This is a very strong assumption on the data, which will be later refined using of the notion of conditional independence in the context of Hidden Markov Random Fields (see Section 11.1.5). Nevertheless, under the previous assumptions, the *probability density function* of the intensity voxel at i -th location can be defined by:

$$P(Y_i) = \sum_{\forall l \in L} P(X_i = l)P(Y_i | X_i = l), \quad (11.7)$$

where $P(X_i = l)$ is the probability of voxel i being of tissue class l , and $P(Y_i | X_i = l)$ is the probability density function of Y_i given the tissue class l . A density function of this form is called *finite mixture* (FM) density. The conditional densities $P(Y_i | X_i = l)$ are called *component densities* and, in this case, they encode the intensity information. The *a priori* probabili-

ties $P(X_i = l)$, are the *mixing parameters* and they encode the spatial information (see Section 11.1.5).

The simplest component density considers that the probability density function for the observed intensity Y_i , given the pure tissue class $X_i = l$, is given by the Gaussian function:

$$P(Y_i | X_i = l, \theta_l) = \frac{1}{\sigma_l \sqrt{2\pi}} e^{-\frac{(Y_i - \mu_l)^2}{2\sigma_l^2}} = f(Y_i | \theta_l), \quad (11.8)$$

where the model parameters $\theta_l = \{\mu_l, \sigma_l\}$ are respectively the mean and the variance of the Gaussian function f . Let us note that this is a good approximation since the noise present in an MRI follows a Rician distribution (Gudbjartsson and Patz 1995; Haacke, Brown et al. 1999) that, at high signal-to-noise ratio, can be approximated by a Gaussian distribution.

In the particular case where the random variables X_i (with $i \in S$) are also assumed to be independent of each other⁶, which means that:

$$P(X_i = l) = \omega_l \quad (11.9)$$

Then, the probability density function for the observed intensities Y_i involved in the image intensity model term of Eq. (11.6) can be written as:

$$P(Y_i | \Theta) = \sum_{l \in L} \omega_l f(Y_i | \theta_l). \quad (11.10)$$

Note that the mixing parameters ω_l are not known in general and must therefore also be included among the unknown parameters. Thus, the mixture density parameter estimation tries to estimate the parameters $\Theta = \{\theta_l, \omega_l\}$ such that:

$$\sum_{l \in L} \omega_l = 1. \quad (11.11)$$

⁶ As it will be shown in Section 11.1.4, this implies that no Markov Random Field model is considered.

11.1.3 The Expectation-Maximization algorithm (EM)

A possible approach to solve the parameter estimation problem in 11.10 is to find the value of Θ using either the *Maximum Likelihood* (ML), or *Maximum Log Likelihood criterion*⁷ (Bilmes 1998):

$$\Theta^* = \arg \max_{\Theta} P(Y | X_i, \Theta) = \arg \max_{\Theta} \log(P(Y | X_i, \Theta)) \quad (11.12)$$

One of the most used methods to solve the maximization problem is the EM algorithm (see Chapter 4). For the particular case of Gaussian distributions, the EM algorithm leads to the following equations:

- *Initialization Step.* Choose the best initialization for $\Theta^{(0)}$.
- *Expectation Step.* Calculate the *a posteriori* probabilities:

$$\hat{P}^{(k)}(X_i = l | y_i, \hat{\Theta}_l^{(k-1)}) = \frac{P(y_i | \hat{\Theta}_l^{(k-1)}) \hat{P}^{(k-1)}(X_i = l)}{\sum_{l \in L} P(y_i | X_i = l, \hat{\Theta}_l^{(k-1)}) \hat{P}^{(k-1)}(X_i = l)} \quad (11.13)$$

- *Maximization Step:*

$$\hat{\omega}_l^{(k)} = \hat{P}^{(k)}(X = l) = \frac{1}{N} \sum_{i \in S} \hat{P}^{(k)}(X_i = l | y_i, \hat{\Theta}_l^{(k-1)}) \quad (11.14)$$

$$\hat{\mu}_l^{(k)} = \frac{\sum_{i \in S} y_i \hat{P}^{(k)}(X_i = l | y_i, \hat{\Theta}_l^{(k-1)})}{\sum_{i \in S} \hat{P}^{(k)}(X_i = l | y_i, \hat{\Theta}_l^{(k-1)})} \quad (11.15)$$

$$(\hat{\sigma}_l^{(k)})^2 = \frac{\sum_{i \in S} (y_i - \hat{\mu}_l^{(k)})^2 \hat{P}^{(k)}(X_i = l | y_i, \hat{\Theta}_l^{(k-1)})}{\sum_{i \in S} \hat{P}^{(k)}(X_i = l | y_i, \hat{\Theta}_l^{(k-1)})} \quad (11.16)$$

⁷ Although both criteria are equivalent, it is usually more convenient to work with the sum of *log* likelihoods (see Chapter 4).

11.1.4 Markov Random Fields (MRF)

In Section 11.1.2, we have made the assumption that all random variables X_i (with $i \in S$) were independent of each other (see Eq. (11.9)). If we want to introduce some relationship between neighbor variables in the random field (which seems a very natural constrain), we have to refine our random field model

We have seen in Chapter 4 that a Markov model (or *Markov chain*) can be used for estimating the probability of a random sequence, based on the assumption that the value taken by i^{th} observation in the sequence only depends on its immediate neighborhood, i.e. on the $(i-1)^{\text{th}}$ observation. Similarly, a *Markov Random Field* (MRF) can be used to estimate the probability of a random image⁸, based on the assumption that the dependency of the value of a given pixel (or voxel) on the values taken by all other pixels in an image, $P(X_i)$, can be reduced to the information contained in a local neighborhood of this pixel.

More formally, let us relate all the pixel (or voxel) locations (*sites*) in a grid S with a *neighborhood system* $N = N_i$ ($i \in S$), where N_i is the set of sites neighboring i , with $i \notin N_i$ and $i \in N_j \Leftrightarrow j \in N_i$. A *random field* X is then said to be an MRF on S with respect to a neighborhood system N if and only if, for all i in S :

$$P(x_i | x_{S-\{i\}}) = P(x_i | x_{N_i}) \quad (11.17)$$

where x_i denotes the actual value of X_i , that is of the random field X at location i ; $x_{S-\{i\}}$ denotes the values of the random field X at all the locations in S except i ; and x_{N_i} the values of the random field X at all locations within the neighborhood of location i . This property of an MRF is known as *Markovianity* respectively.

According to the Hammersley-Clifford theorem, an MRF can be equivalently characterized by a Gibbs distribution,

$$P(\mathbf{x}) = \frac{e^{-U(\mathbf{x}, \beta)}}{Z} \quad (11.18)$$

⁸ The definitions here proposed can be related to 2D or 3D images. In the case of 2D images, the single elements are pixels whereas in the case of 3D images, they are voxels.

where \mathbf{x} represents a simultaneous configuration of the random variables, X_i ; that is, a given label map (see Section 11.1.1).

The expression in Eq. (11.18) has several *free* parameters to be determined: the *normalization factor* Z , the *spatial* parameter β , and the *energy function* U . Let us briefly discuss how these parameters can be determined in the particular framework of image segmentation. We refer the interested reader to (Li 2001) for further details.

The energy function U

Given the previously commented Hammersley-Clifford theorem, we can include a specific definition of the spatial distribution model in Eq. (11.6). This way, the maximization proposed in Eq. (11.6) can be formulated in terms of the energy function:

$$\log[P(\mathbf{y} | \mathbf{x}, \Theta)P(\mathbf{x})] = \log P(\mathbf{y} | \mathbf{x}, \Theta) - U(\mathbf{x}, \beta) + \text{const} \quad (11.19)$$

The definition of the energy function is arbitrary and several choices for $U(\mathbf{x})$ can be found in the literature related to image segmentation. A general expression for the energy function can be denoted by:

$$U(\mathbf{x}, \beta) = \sum_{\forall i \in S} \left(V_i(x_i) + \frac{\beta}{2} \sum_{j \in N_i} V_{ij}(x_i, x_j) \right) \quad (11.20)$$

This is known as the *Potts model* with an *external field*, $V_i(x_i)$, that weights the relative importance of the various classes present in the image (see, for instance, (Van Leemput, Maes et al. 2003)). However, the use of an external field includes additional parameter estimation, thus this model is less used in image segmentation. Instead, a simplified Potts model with no external energy, $V_i(x_i) = 0$, is used. Then, only the local spatial transitions $V_{ij}(x_i, x_j)$ are taken into account and all the classes in the label image are considered equally probable. A common way to define them is, for instance, as:

$$V_{ij}(x_i, x_j) = -\delta(x_i, x_j) = \begin{cases} -1, & \text{if } x_i = x_j \\ 0, & \text{otherwise.} \end{cases} \quad (11.21)$$

Intuitively, the equation above encourages one voxel to be classified as the tissue to which most of its neighbors belong. That is, it favors configurations where homogenous classes are present in the image.

The spatial parameter β

Given the energy function defined in Eq. (11.20) and (11.21), the maximization proposed in Eq. (11.6) can be formulated as a minimization problem:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in X} [-\log P(\mathbf{y} | \mathbf{x}, \Theta) + \beta U(\mathbf{x})] \quad (11.22)$$

As it can be observed, the *spatial* value of β controls the influence of the energy factor due to the spatial prior, $U(\mathbf{x})$, over the factor due to the intensity, $-\log P(\mathbf{y} | \mathbf{x}, \Theta)$. Note that its influence on the final segmentation is important. For instance, $\beta = 0$ corresponds to a uniform distribution over the possible states, that is, the maximization is done only on the conditional distribution of the observed data $P(\mathbf{y} | \mathbf{x}, \Theta)$. On the contrary, if the spatial information is dominant over the intensity information, that is if $\beta \rightarrow \infty$, MAP tends to classify all voxels into a single class.

The value of β can be estimated by ML estimation. However, many problems arise due to the complexity of MRF models and alternative approximations have to be done. Commonly, this is done by simulated annealing, such as Monte-Carlo Simulations (Geman and Geman 1984), or by maximum pseudo-likelihood approaches (Besag J. 1974). The β parameter can be also determined *arbitrarily* by gradually increasing its value over the algorithm iterations. Here, the value of β will be fixed empirically and we will study its influence on the final segmentation in Section 11.2.5.

The normalization factor Z

The normalization factor of Gibbs distribution is theoretically defined as:

$$Z(U) = \sum_{\mathbf{x} \in X} e^{-U(\mathbf{x}, \beta)} \quad (11.23)$$

but this definition implies a high computational cost or it is even intractable since the sum among all the possible configurations of X is usually not known (Geman S. and Geman D. 1984). Note also its dependence on the

definition of the energy function U . Nevertheless, in optimization problems, the solution does not depend on the normalization factor since it is a constant value shared by all possible solutions. Therefore, commonly, the exact computation of the normalization factor is circumvented.

11.1.5 Hidden Markov Random Fields (HMRF)

In Section 11.1.2 and 11.1.3, we have examined how GMMs can be used for estimating the image intensity model, while assuming that Y_i are independent and identically distributed. In this Section, we will introduce some form of dependency among variables Y_i through the use of an underlying, hidden MRF for spatial distribution modeling.

The theory of Hidden Markov Random Field (HMRF) models is derived from Hidden Markov Models (HMMs), as seen in Chapter 4. HMMs are defined as stochastic processes generated by a Markov chain whose state sequence X cannot be observed directly, but only through a sequence of observations Y . As a result, HMMs model a random sequence of observations Y with two sets of probabilities: the emission and the transition probabilities, which respectively account for the local and sequential randomness of the observations. Here we consider a special case since, instead of a Markov chain, an MRF is used as the underlying stochastic process; that is, the sequential correlation between observations will be replaced by the spatial correlation between voxels.

The HMM concept can then be adapted to MRFs, leading to the notion of Hidden MRF (HMRF). As HMMs, HMRFs are defined with respect to a pair of random variable families (X, Y) where X accounts for spatial randomness, while Y accounts for local randomness. In summary, a HMRF model is characterized by the following:

- **Hidden Random Field:** $X = \{X_i, i \in S\}$ is an underlying MRF assuming values in a finite state space L with probability distribution as defined in the Gibbs equation (11.18).
- **Observable Random Field:** $Y = \{Y_i, i \in S\}$ is a random field with a finite state space D . Given any particular configuration, $\mathbf{x} \in L^{\mathbb{N}}$, every Y_i follows a known conditional probability distribution $P(Y_i | X = \mathbf{x})$ of the same functional form $f(Y_i | \theta_{\mathbf{x}})$, where $\theta_{\mathbf{x}}$ are the involved parameters. This distribution is called *emission probability function* and, Y is also referred to as the *emitted random field*.
- **Conditional Independence:** For any $\mathbf{x} \in L^{\mathbb{N}}$, the random variables Y_i are supposed to be independent, which means that :

$$P(\mathbf{y} | \mathbf{x}) = \prod_{i \in S} P(y_i | x_i) \quad (11.24)$$

So now, as in Section 11.1.2, it is possible to compute the probability density function for the observed intensities Y_i involved in the image intensity model term of Eq. (11.6). In the HMRF case, the probability density function of Y_i dependent on the parameter set Θ and the values in the associated neighborhood of the hidden random field, x_{N_i} , is

$$\begin{aligned} P(Y_i | X_{N_i}, \Theta) &= \sum_{l \in L} P(Y_i, X_i = l | x_{N_i}, \Theta_l) \\ &= \sum_{l \in L} P(X_i = l | x_{N_i}) P(Y_i | \theta_l). \end{aligned} \quad (11.25)$$

where $\Theta = \{\theta_{x_i}, x_i \in \ell\}$. This is again a *finite mixture* (FM) density as in Eq. (11.10) where, now, the *a priori* probabilities $P(X_i = l | x_{N_i})$ are the *mixing parameters*.

11.1.6 Gaussian Hidden Markov Random Field Model (GHMRF)

In the Gaussian case we can assume:

$$P(Y_i | \theta_l) = \frac{1}{\sigma_l \sqrt{2\pi}} e^{\frac{-(Y_i - \mu_l)^2}{2\sigma_l^2}} = f(Y_i | \theta_l), \quad (11.26)$$

where, $f(Y_i | \theta_{x_i})$, is a Gaussian distribution defined by $\theta_l = \{\mu_l, \sigma_l\}$. The probability density function of the intensity voxels can now be written as function of the parameter set Θ and of the voxel neighborhood X_{N_i} as:

$$P(Y_i | X_{N_i}, \Theta) = \sum_{l \in L} P(X_i = l | x_{N_i}) f(Y_i | \theta_l). \quad (11.27)$$

where $P(X_i = l | x_{N_i})$ represents the locally dependent probability of the tissue class l . Note that in Section 11.1.2 we simplified the analogous probability: $P(X_i = l) = P(X = l) = \omega_l$. Here, the independence assumption is not valid any more.

To solve the parameter estimation problem, an adapted version of the EM algorithm presented above, called the HMRF-EM, is used (Zhang, Brady et al. 2001). The update equations for the $\theta_l = \{\mu_l, \sigma_l\}$ parameters are actually the same update equations as for the *Finite Gaussian Mixture Model (FGMM)* (Equations 11.15 and 11.16) except that

$$\hat{P}^{(k)}(X_i = l | y_i, \hat{\Theta}_l^{(k-1)}) = \frac{P(y_i | \hat{\Theta}_l^{(k-1)}) \hat{P}^{(k-1)}(X_i = l | x_{N_i})}{\sum_{l \in L} P(y_i | X_i = l, \hat{\Theta}_l^{(k-1)}) \hat{P}^{(k-1)}(X_i = l | x_{N_i})} \quad (11.28)$$

Note that the calculation of $\hat{P}^{(k-1)}(X_i = l | x_{N_i})$ involves a previous estimation of the class labels, X_i ; that is, the classification step. Typically, an initial classification is computed using the GMM approach, thus without the HMRF model (see Section 11.2.4).

11.2 MATLAB proof of concept

11.2.1 3D data visualization

As previously commented, we will do a *longitudinal study*, that is, we will compare two MR images from the same patient acquired in a year interval (I1 in 2000 and I2 in 2001, see Fig. 11.2). Studies on patterns of brain atrophy in medical images differ however from the proof-of-concept presented here. Usually, such studies are performed statistically, in comparison to a probabilistic pattern that represents the normal anatomy, that is, a *transversal study*. Nevertheless, the statistical classification framework presented here remains valid.

The image data used here are 3D MR images of the human brain. They are T1-weighted⁹ images of $186 \times 186 \times 121$ voxels of dimension $1 \text{ mm} \times 1 \text{ mm} \times 1.3 \text{ mm}$. Voxel type is *unsigned short* integers coded in 16 bits, that is voxel values can vary from 0 to 65535.

⁹ Different image contrast can be generated in the MR imaging process; the most often used is T1-weighted but also T2 and Proton Density (Haacke et al. 1999).

MATLAB provides several techniques for visualizing 3D scalar volume data¹⁰. One of them is the `contourslice` function:

```
load ImageVolume.mat
phandles=contourslice(D,[],[45,64,78,90,100,110,120,127,135,142,150,160,175,190,200,210,220,228],[60,65,70,75,80,85],7);
view(-169,-54); colormap(gray)
```

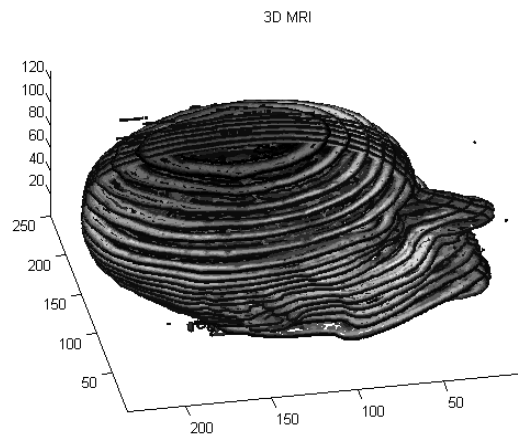


Fig. 11.4 Visualization of 3D MRI data

The resulting volume is shown in Fig. 11.4. The *view angle* can be changed by using *Tools* and *Rotate 3D* in the figure menu. However, medical doctors usually visualize such data using the 3 main views, namely the *Axial*, *Coronal* and *Sagittal* views. After loading the volume into the MATLAB *Workspace* an *Axial* matrix that contains the image intensities appears. Its dimensions are:

```
load Image1_voi.mat
size(Axial)
ans= 186 186 121
```

Let us have a look at the three views of the *Axial* matrix:

```
subplot(1,3,1); imagesc(squeeze(Axial(90,:,:)));
subplot(1,3,2); imagesc(squeeze(Axial(:,65,:)));
subplot(1,3,3); imagesc(squeeze(Axial(:, :, 60))');;
```

¹⁰ Search for ‘Techniques for Visualizing Scalar Volume Data’ in the Help menu of MATLAB.

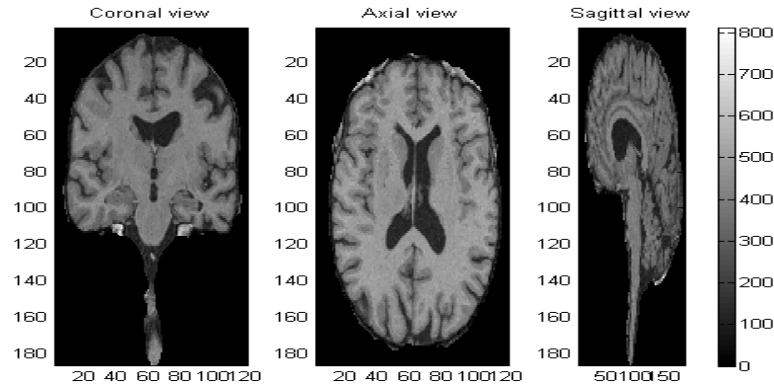


Fig. 11.5 Coronal, Axial and Sagittal views of a 3D MR brain image at time 1

Thus the same volume as in Fig. 11.4 is seen here from three different 2D views: from the front of the head (Coronal), from the top of the head (Axial) and from a side of the head (Sagittal). Note that only the brain is presented, since the skull has been removed¹¹. The color map is as follows: dark gray is Cerebrospinal Fluid, or CSF; middle gray is Gray Matter, or GM; and light gray is White Matter, or WM. Note that few patches of the skull are still present.

Now, we can run the same code for the image at time 2 (see Fig. 11.6):

```
load(Image2_voi.mat);
```

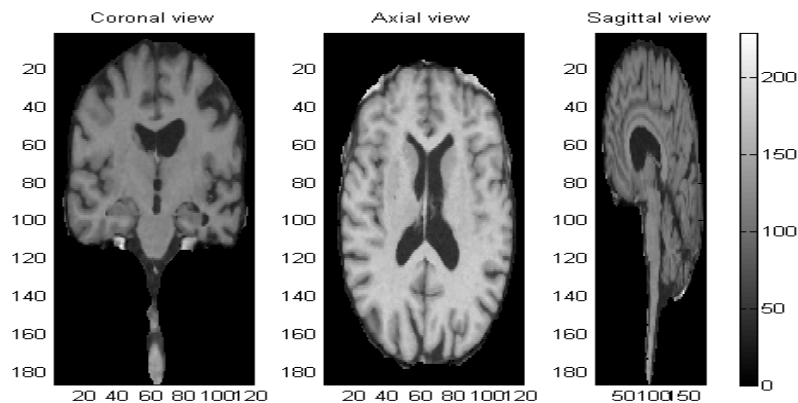


Fig. 11.6 Coronal, Axial and Sagittal views of a 3D MR brain image at time 2

¹¹ Several techniques can be used to remove skull and other non-brain tissues from brain. Here morphological operations (dilations and closings) have been applied.

Note that both images look very similar: they show the same brain, both have the same size and there is a spatial correspondence (they have been already rigidly matched); that is, they have a point to point correspondence (see Section 11.3). However, as the *colorbar* on the right side shows, there is a difference in the intensity level range. Thus we propose hereafter to study the image histogram.

11.2.2 Image histogram

The function involved in the computation of the image histogram is:

```
[H,X,NbrBin,BinSize]=histogram(Axial,loffset,roffset);
```

The outputs are the values of the image histogram *H*, the bins vector *X*, the total number of bins, *NbrBin*, and the bin size is *BinSize*. *loffset* and *roffset* correspond respectively to the left and right offsets that define the start and end of the image intensities considered in the computation. Note that images in Fig. 11.5 and Fig. 11.6 present large black areas (0 grey level) which do not provide any information. To remove these areas from the statistical analysis, we set the left offset to 1. In turn, the right offset is set to 0. Results are shown in Fig. 11.7:

```
loffset=1;  
roffset=0;  
  
load Image1_voi.mat  
[H,X,NbrBin,BinSize]=histogram(Axial,loffset,roffset);  
  
load Image2_voi.mat  
[H,X,NbrBin,BinSize]=histogram(Axial,loffset,roffset);
```

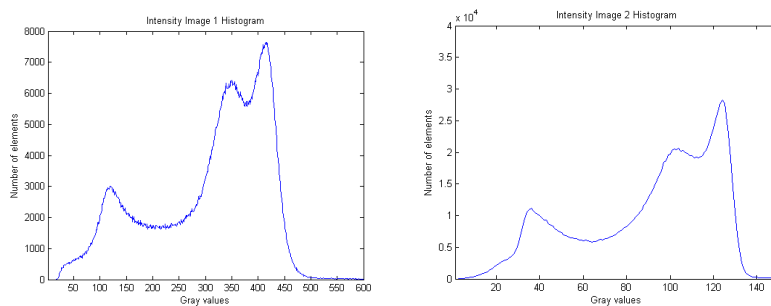


Fig. 11.7 Image histogram, left is image 1 (year 2000) and right is image 2 (year 2001)

As expected, intensity range values are different (as already seen with the *colorbar* in Fig. 11.5 and Fig. 11.6). Three peaks can be observed: the first one is the darkest value and corresponds to CSF, the second one represents GM and the third one corresponds to WM. The current number of bins used in the representation is

```
NbrBin
ans = 1179; % For Image 1
ans = 301; % For Image 2
```

Our `histogram` function calls the `histc` MATLAB function:

```
H=histc(Axial(:),X);
```

This function counts the number of values in `Axial` that fall between the elements in the `X` vector of bins (which must contain monotonically non-decreasing values). Then, `H` is a vector of length `length(X)` containing these counts.

We can change the range of bins `X` interactively¹² by choosing a different left and right offset:

```
lims=getpts;
disp('')
loffset=floor(lims(1))
roffset=floor(NbrBin-lims(2))

loffset = 23
roffset = 696
```

The number of bins can be modified by not only varying the extremes of vector `X` but also by changing the bin size. This is implemented by changing the `step` value in the `ASP_brain_segmentation_histogram.m` function:

```
if roffset==0
    X=loffset:step:n;
else
    X=loffset:step:(n-roffset);
end
```

This way the histogram resolution can be modified as seen in Fig. 11.8 where `step` value has been set to 0.1 (left plot) and 40 (right plot):

¹² The `getpts` function allows you to select points from the current figure with the mouse (left-button) and select last point with the right-button.

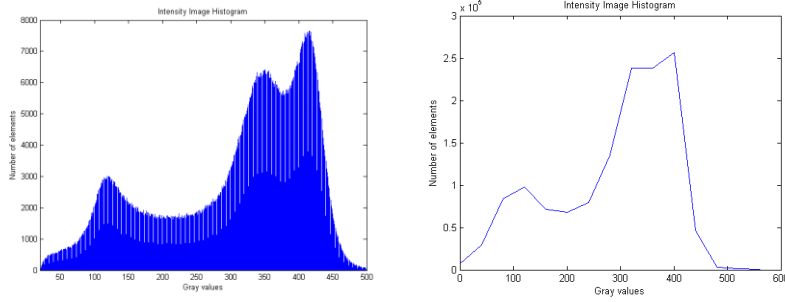


Fig. 11.8 Image histograms using step=0.1 (left) and step=40 (right)

The selection of the number and the size of the bins in the previous example is only a matter of visualization. However, these parameters can have an influence in the computation time and precision of some optimization problems where image histograms are computed very often. We will show in the next Section the influence of the bin size on the Gaussian fitting optimization problem.

In this work, we will approximate the probability density function (PDF) of the intensity values by the normalized image histogram. The normalization of the image histogram nH is computed as follows:

```
NbrSamples=sum(sum(H));
nH=H*1/NbrSamples;
```

11.2.3 Gaussian Mixture Model (GMM)

As presented in Section 11.1.3, to solve the parameter estimation of the Gaussian Mixture Model we use the Expectation Maximization (EM) algorithm. Such a model assumes that the probability density function of the image intensity of every class follows a Gaussian distribution. Before discussing the estimation problem, let us have a look at the MATLAB functions involved in the generation of the Gaussian probability densities:

```
y = exp(-0.5*(x-mu).*(x-mu)/(sigma*sigma))...
/(sqrt(2*pi)*sigma);
```

This is packed into the `ASP_brain_segmentation_gauss.m` function and corresponds to Eq. (11.8). In this equation, x is the image intensity values and μ and σ must be estimated. In fact, we are looking for the parameters of a weighted sum of Gaussian distributions that better fit the image intensity histogram (likelihood estimation). `NbrComp` is the

number of mixtures used for the fitting which, in our case, is equal to three since three classes are used: CSF, GM and WM.

```
for i=1:NbrComp
    for j=1:length(X)
        G(i,j)=Theta(i,3).*gauss(X(j),Theta(i,1),Theta(i,2));
    end
end
```

G is a $\text{NbrComp} \times \text{NbrBin}$ matrix, in which $\text{Theta}(:,1)$ are the mean, $\text{Theta}(:,2)$ are the standard deviations and $\text{Theta}(:,3)$ are the weights.

Initial values of means, standard deviations and weights must be set. Remember that the EM algorithm is sensitive to initial parameters and, particularly, to initial mean values (see Chapter 4 for an example). We can modify the initial parameters by manually selecting the mean values and, this way, observe the convergence behavior. Let us first set 3 mean values on the left side of the image histogram:

```
mus=[130 220 330];
s=size(mus);
NbrComp=s(1);
```

Note that mean values can be also selected by clicking on the image histogram; the corresponding MATLAB code is provided in the main script, `ASP_brain_segmentation.m`.

Then, the standard deviation and the initial weight are usually set as:

```
for i=1:NbrComp
    sig(i,1)=5.0;
    pw(i,1)=1/NbrComp;
end
```

That is, standard deviation can be set to a small but non-zero value and weights are all set equally. As a matter of fact, the EM algorithm is not as sensitive to these parameters as to the mean values. Finally, initial parameters are saved in Theta :

```
% compose Theta
Theta=cat(2,mus,sig,pw);
```

```
Theta =
    130    5.0000    0.3333
    220    5.0000    0.3333
    330    5.0000    0.3333
```

Now, the EM algorithm can be applied:

```
[S,e,Theta,fitper,G]=segmentem(Axial, loffset, roffset,
Theta);
```

This function computes the following output parameters: the new estimated **Theta** values, the corresponding Gaussian distributions **G**, the squared error **fitper** between the image histogram and the fit with the Gaussian mixture, the threshold boundaries **e** between each mixture, and the estimated classification **S** obtained by applying a threshold to the intensity values with **e**. The result is plotted in Fig. 11.9. In it, three different types of functions can be observed. The original histogram is presented in dot line, whereas the approximated histogram is in solid line. This approximated histogram is the sum of the three Gaussian histograms that have been estimated for each class, which are presented in dashed lines. The vertical lines represent the two different thresholds that have been finally obtained and applied to perform the classification.

```

fitper = 84.6362
Estimated means and variance:
u1 =
    126.3722
    225.6552
    368.8622
s1 =
    42.2332
    35.0064
    55.6226

Estimated thresholds for Bayesian decision:
e = 0    190    256

```

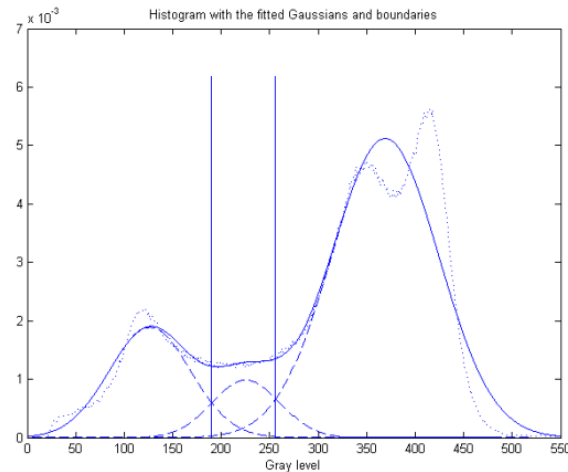


Fig. 11.9 Fitting of the image histogram: 3 Gaussians are used. Initial means are 130, 220, and 330 and bin size is 1

Let us now visualize the image segmentation S after thresholding the input image using the e values:

```
figure; clf;
subplot(1,3,1); imagesc(squeeze(S(90,:,:)));
iptsetpref('ImshowBorder','tight');
title('Coronal view'); hold on;
subplot(1,3,2); imagesc(squeeze(S(:,65,:)));
iptsetpref('ImshowBorder','tight');
title('Axial view'); hold on;
subplot(1,3,3); imagesc(squeeze(S(:,:,60)));
axis fill;
iptsetpref('ImshowBorder','tight');
colorbar; title('Sagittal view');
```

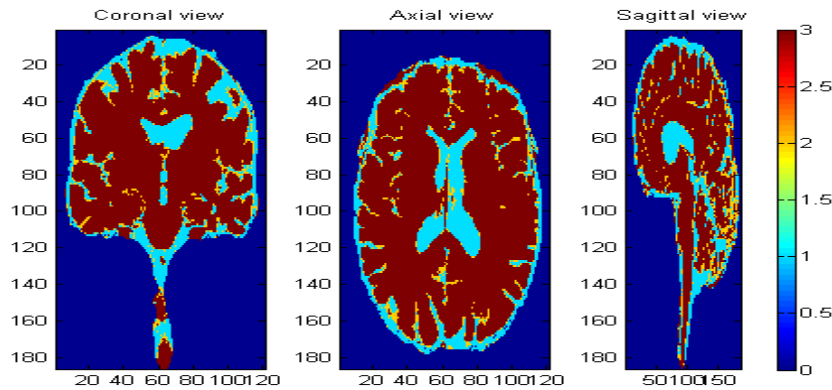


Fig. 11.10 Bayesian classification in 3 classes: 0 is background, 1 is CSF, 2 is a thin transition class, and 3 is GM and WM all together. Initial means are 130, 220, and 330 and bin size is 1.

Obviously, the resulting image classification of Fig. 11.10 is wrong since the Gaussian fitting has failed. GM and WM have been mixed together into a single class.

In order to further analyze the sensitivity of the EM algorithm with respect to the initialization, let us now use a set of initial mean values closer to the three peaks of the image histogram, keeping the initial values of standard deviation and weights as in the previous example:

```
mus = [115 345 410]';
```

```
theta =
```

115	5.0000	0.3333
345	5.0000	0.3333
410	5.0000	0.3333

This time, the result of `ASP_brain_segmentation_segmentem.m` is

```

fitper = 87.3078
Estimated mean and variance:
u1 =
138.6997
347.4169
414.2959
s1 =
49.6081
63.1638
17.1897
Estimated thresholds for Bayesian decision:
e =
0 219 401

```

This result is presented in Fig. 11.11 following the same layout that has been used in Fig. 11.9.

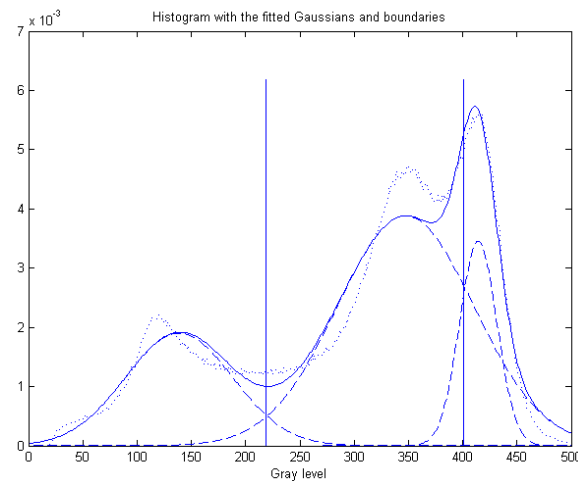


Fig. 11.11 Fitting of the image histogram: 3 Gaussians are used. Initial means are 115, 345, and 410 and bin size is 1.

The fitting and classification results are now more satisfactory: classes converged to the underlying anatomical tissues. The resulting image classification in Fig. 11.12 shows a good convergence of the algorithm. Unfortunately it is quite noisy; note the spurious points that are misclassified as gray matter in the interior of white matter. This is due to the Bayesian decision, which is taken based on the intensity gray level information only. Better results can be obtained by including local spatial information about

the neighboring voxels. This can be done applying the MRF theory as seen previously in the theory Section 11.1.4 and presented hereafter.

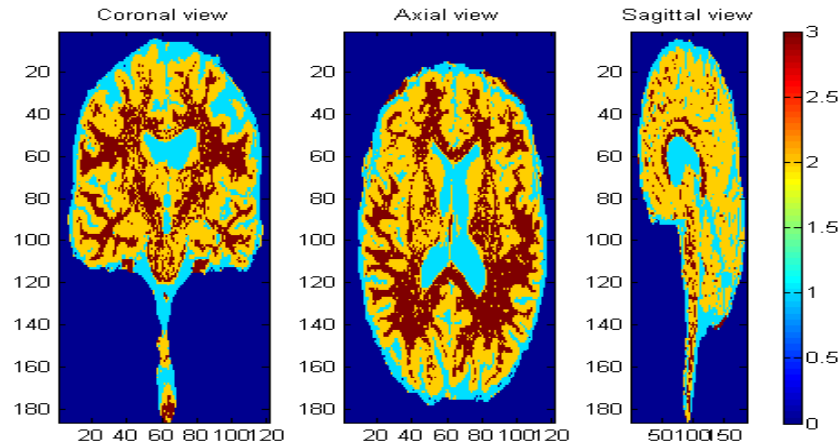


Fig. 11.12 Bayesian classification in 3 classes: 0 is background (in dark blue), 1 is CSF (in light blue), 2 is GM (in yellow), and 3 is WM (in dark red). Initial means are 115, 345, and 410.

As mentioned in the previous Section, the result of the Gaussian fitting also depends on the selected number and size of bins. Logically, if we change the reference image histogram the final fitting will change. Let us illustrate this using a bin size of 20 instead of 1 and keeping the same initialization as in the previous case (see Fig. 11.13):

```
mus = [115 345 410]';

fitper = 75.0771

Estimated means and variance:
u1 =
    130.5112
    291.7118
    390.3842
s1 =
    38.9105
    98.6497
    42.1912

Estimated thresholds for Bayesian decision:
e =
     0    161    321
```

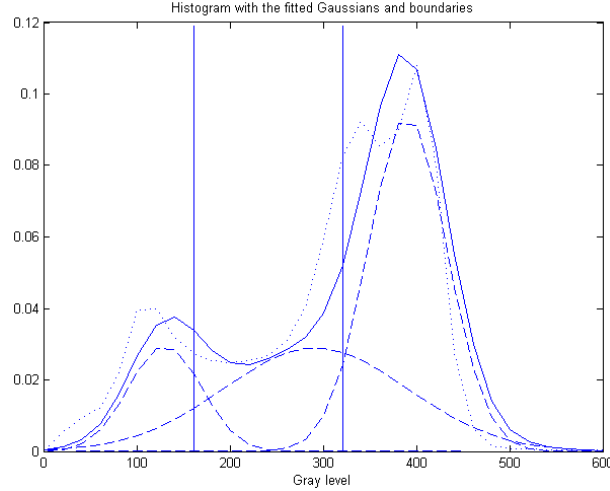


Fig. 11.13 Fitting of the image histogram using a different bin size to compute the image histogram. Initial means are 115, 345, and 410 and bin size is 20.

In terms of computational time, the smaller the number of bins, the faster the fitting. However, as illustrated in this example, there is a compromise since the use of too few bins leads to an inaccurate histogram estimate. As it can be seen, the selected thresholds are not as accurate as in the previous case (see Fig. 11.12).

Moreover, in that particular case, the use of a too small number of bins presents another effect. Note that the threshold obtained between class 1 (CSF) and class 2 (GM) is not coincident with the intersection of the Gaussian functions representing their individual density functions. This is due to the rough quantification into a too small number of bins: the theoretical value (which is the intersection point between both density functions) is approximated by the center of the closest bin.

11.2.4 Hidden Gaussian Mixture Model

The strategy underlying the HMRF-EM algorithm consists in applying iteratively the following two steps:

1. Estimate the image labeling, \mathbf{x} , given the current model θ and then use it to form the complete data set \mathbf{x}, \mathbf{y} (\mathbf{y} being the intensity image observation). The labeled image is initially estimated only using GMM

(Section 11.2.3). Then, solve the following minimization problem (instead of maximizing the posterior probability):

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in X} [-\log P(\mathbf{y} | \mathbf{x}, \Theta) + \beta U(\mathbf{x})] \quad (11.29)$$

The MATLAB functions involved are:

```
nL=daLoop(nL,Axial,u1,s1,NbrComp,beta);
```

which computes the new classified image (**nL**) as in Eq. (11.22) with the current estimation of mean (**u1**) and variance (**s1**). It includes two functions for the calculation of the two energies:

```
calcux(mu(1),sigma(1),Image);
```

```
calcux_3class(L,Sx,Sy,Sz,1);
```

where **1** is the current label, **Sx**, **Sy** and **Sz** are the voxel dimensions (1 mm, 1 mm and 1.3 mm, respectively) and **L** is the current classified image.

2. Estimate a new θ by maximizing the expectation of the complete data log likelihood, $E[\log P(\mathbf{x}, \mathbf{y} | \theta)]$. That is, re-estimate the mean and variance vectors as described in Section 11.1.6 and update the equations for mean and variance vectors in the EM algorithm.

The MATLAB function involved is

```
[u1,s1]=calcu1s1(Axial, nL, nH, X, u1, s1,NbrComp);
```

The two parameters to set in the HMRF-EM algorithm are the number of iterations and the value of β (see Eq. (11.22)) that weights the energy $U(\mathbf{x})$ with respect to $-\log P(\mathbf{y} | \mathbf{x}, \Theta)$. The loop is:

```
nit=5;  
beta=0.6;  
for ii=1:nit  
    nL=daLoop(nL,Axial,u1,s1,NbrComp,beta);  
    [u1,s1]=calcu1s1(Axial, nL, nH, X, u1, s1,NbrComp);  
end
```

The final label map after 5 MRF iterations is shown in Fig. 11.14. Note that it is much less noisy than the label map shown in Fig. 11.12.

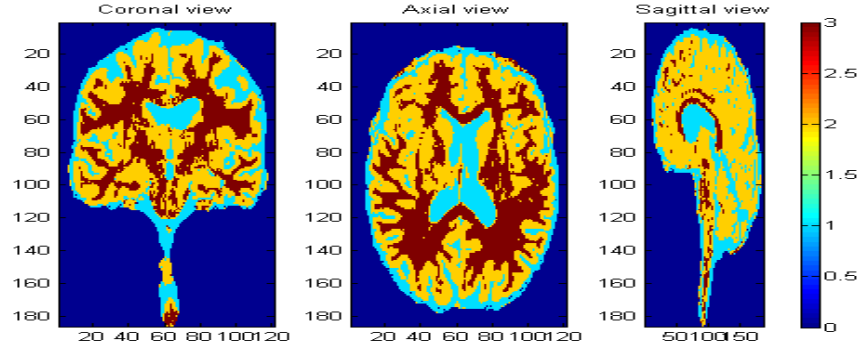


Fig. 11.14 HMRF classification in 3 classes after 5 iterations and β set to 0.6.

11.2.5 Influence of the spatial parameter

As mentioned in Section 11.1.4, the β parameter is set here experimentally. In this Section we will analyze its influence on the final segmentation. Let us first observe the HMRF classification with $\beta = 0.1$, $\beta = 1.2$, and $\beta = 10$ after 60 iterations (see Fig. 11.15).

Let us now study the HMRF convergence. For this purpose we compute the percentage of voxels (related to the total number of voxels of the image, without considering the background) that changed of assigned label, between two consecutive iterations, during 20 iterations (see Fig. 11.16). This will give us an idea about the *speed* of convergence to a local optimal solution for the MAP decision.

The corresponding MATLAB code is included in the main loop of HRMF:

```

nit=60;
beta=10;

nLold=nL;
change=zeros(nit,1);

for ii=1:nit
    nL=daLoop(nL,Axial,u1,s1,NbrComp,beta);

    Diff=(nLold-nL);
    Change(ii)=length(find(Diff))./length(find(nL))*100;
    nLold=nL;

    [u1,s1]=calculs1(Axial, nL, nH, X, u1, s1,NbrComp);
end

```

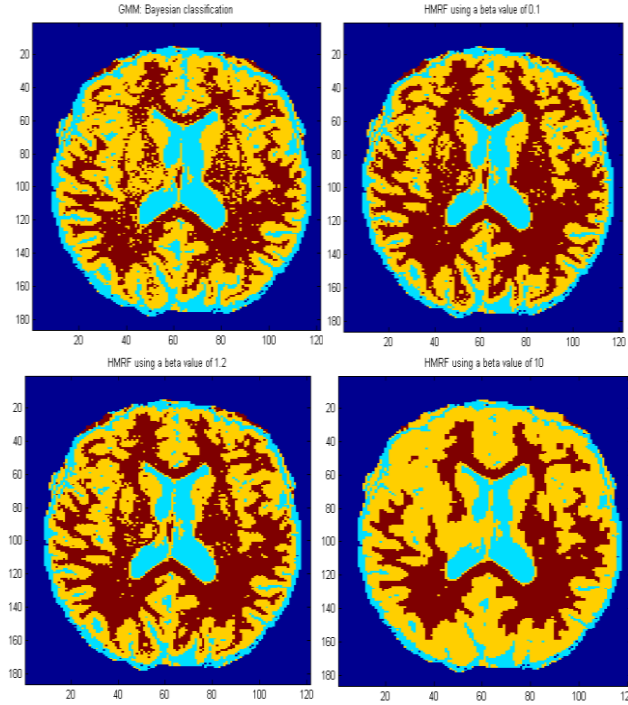


Fig. 11.15 Segmentation after 60 HMRF iterations. Top left: Bayesian classification without HMRF; Top right: HMRF classification with $\beta = 0.1$. Bottom left: HMRF classification with $\beta = 1.2$; Bottom right: same with $\beta = 10$.

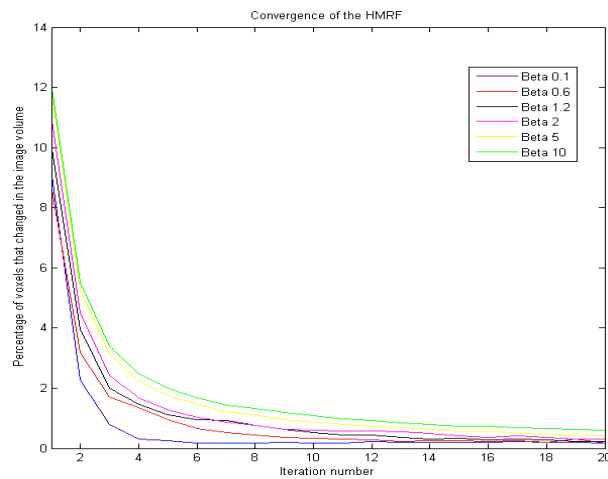


Fig. 11.16 HMRF convergence study

As seen in Fig. 11.16, the larger β is, the slower the convergence to a MAP solution is. There is thus a compromise between speed (to reach a solution) and quality of the final segmentation. In this work we will keep the values of $\beta = 0.6$ and $n_{it} = 5$ iterations. This choice is still arbitrary.

11.2.6 Localization and quantification of brain degeneration

Our goal here is to localize and quantify the GM degeneration in the previous example. Globally, studies on patterns of brain atrophy in medical images are done mainly by two different approaches. The first consists of the manual detection and classification of a region of interest (ROI), leading to a time consuming and subjective procedure (Galton, Patterson et al. 2001). The second, and most used approach, is the *voxel-based morphometry* (VBM), usually having the following steps (Ashburner and Friston 2000):

- Normalization of all the subjects into the same space,
- Gray matter extraction and smoothing of the normalized segmented images,
- Statistical analysis of the differences between *a priori* reference data and the subjects.

The method proposed here is in fact based on the VBM theory. In our study, the probabilistic reference is not available so the evolving degeneration is only studied from the sequence images; that is, a so-called *longitudinal* study is performed.

Our interest is not in the final tissue classification (label map), but in the tissue probability maps; that is, functions (images in our case) in which each point (pixel) has associated the likelihood of belonging to a given tissue. Particularly, as GM degeneration is studied, cerebrospinal fluid (CSF) and GM posterior probability maps are retained from the classification step. Working with probability maps will allow us to be more precise in the computation of average concentration maps.

The MATLAB functions involved are:

```
ProbMaps(Axial, nL, u1, s1, NbrComp, filenameMRI(1:end-4));
```

This function outputs the posterior probability maps into .raw files for every tissue class:

```
%write output files  
fname=strcat(FileName, '_Pcsf.raw');  
fid = fopen(fname, 'wb');  
fwrite(fid, A*255, 'float');
```

fclose(fid)

All probability maps are re-scaled to gray levels in $[0, 255]$ before being stored in *.raw* files. These probability maps are in fact computed as:

```
ptly(1,:,:,:)=calcptly(Image, L, 1, nH, X, mu(1), sigma(1),
NbrComp);
```

which implements the equation described in Eq. (11.28).

Then, a smoothing of the tissue probability maps by a large Gaussian kernel is applied. This way, each voxel of the tissue probability map represents the tissue concentration within a given region defined by the standard deviation of the Gaussian filter being used.

In order to determine the standard deviation, morphometry theory, as described in (Ashburner and Friston 2000), suggests a value similar to the size of the *changes* that are expected. For the particular case of the image sequence under study, the value of $\sigma = 11$ mm is used. The resulting concentration maps of CSF and GM at time 1 are shown in Fig. 11.17 and Fig. 11.18, respectively.

```
Ccsf_1=smooth_3DGauss('Image1_voi_Pcsf.raw',s(1),s(2),s(3),
'float',7,11);
Cgm_1=smooth_3DGauss('Image1_voi_Pgm.raw',s(1),s(2),s(3),
'float',7,11);
Ccsf_2=smooth_3DGauss('Image2_voi_Pcsf.raw',s(1),s(2),s(3),
'float',7,11);
Cgm_2=smooth_3DGauss('Image2_voi_Pgm.raw',s(1),s(2),s(3),
'float',7,11);
```

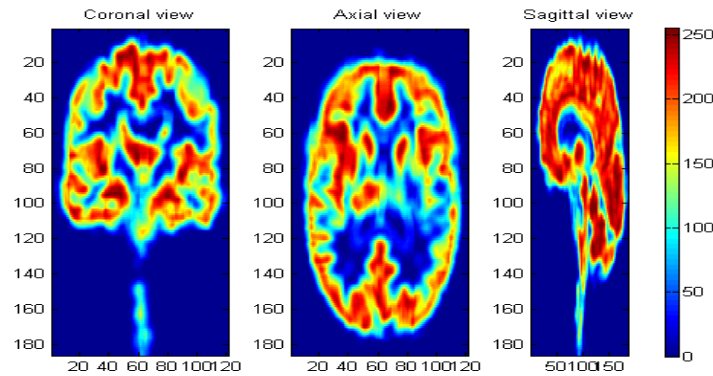


Fig. 11.17 Smoothed probability maps for the GM class.

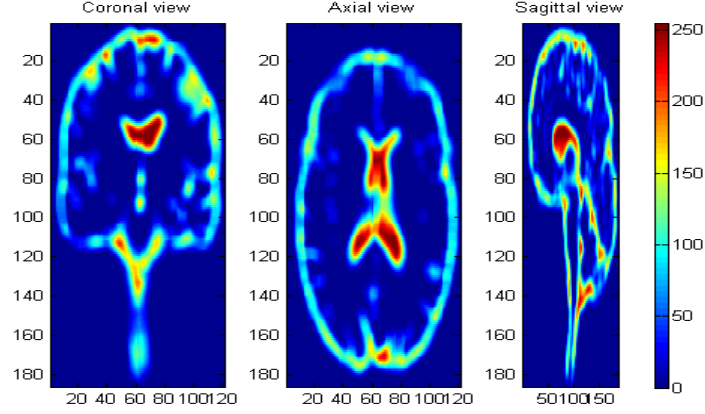


Fig. 11.18 Smoothed probability maps for the CSF class.

Let us define the GM degeneration map as

$$D_{12} = \prod_l |C_1^l - C_2^l| \quad (11.30)$$

where C_t^l is the concentration map of tissue l at time t , for $l \in \{CSF, GM\}$, $t \in \{1, 2\}$, and D_{12} shows the regions with more probable gray matter loss between images 1 and 2. The corresponding MATLAB code is:

```
D12=abs(Ccsf_1-Ccsf_2).*abs(Cgm_1-Cgm_2);
imagesc(squeeze(D12(:,99,:)));
imagesc(squeeze(D12(84,:,:)));
```

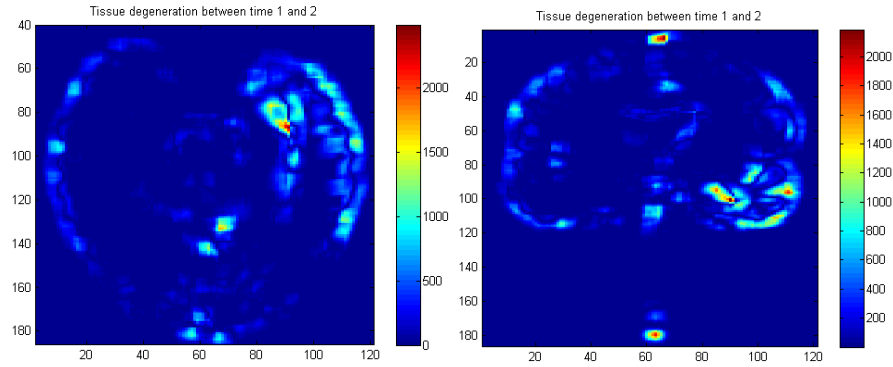


Fig. 11.19 Detected degeneration areas: Left is an axial view, right is the coronal view. Highest values represent the highest degenerated regions

The obtained degeneration maps (Fig. 11.19) may contain some regions that actually do not correspond to a real GM degeneration (for instance due to registration errors in the cortex or trunk area). However, the most important regions of degeneration (ROD) can be isolated by applying a simple threshold to the previous images. The result is presented in Fig. 11.20 where it can be observed that degeneration can now be easily quantified.

```
volume_roi_1=sum(cgm_1(D12>500))
```

```
7.4180e+006
```

```
volume_roi_2=sum(cgm_2(D12>500))
```

```
6.1643e+006
```

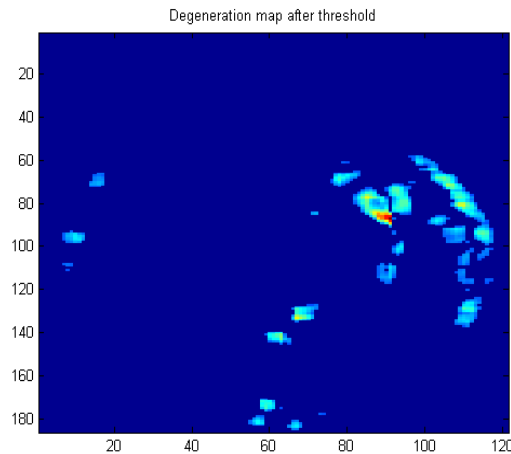


Fig. 11.20 Degeneration areas after thresholding

```
degeneration_roi=(volume_roi_1-volume_roi_2)/volume_roi_1*100
```

```
16.9012
```

Medical diagnosis of this patient is GM degeneration in the entorhinal region. This corresponds approximately to the region shown in Fig. 11.21:

```
ROD = D12(45:115,70:120,70:121);
```

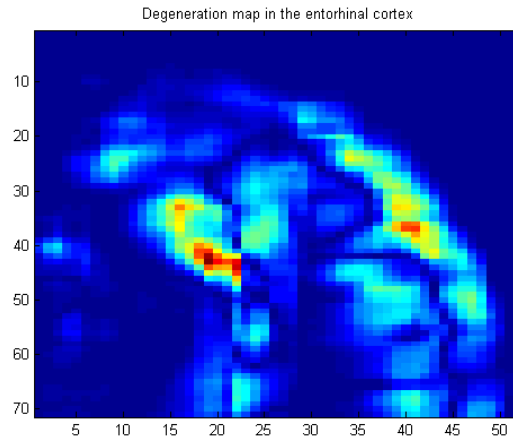


Fig. 11.21 Degeneration map in the entorhinal cortex

Once the candidates to define the GM degeneration regions have been detected, degeneration can be quantified more precisely within these regions:

```
volume_roi_1=sum(Cgm_1(ROD~=0))  
2.7792e+004  
volume_roi_2=sum(Cgm_2(ROD~=0))  
2.3326e+004  
degeneration_roi=(volume_roi_1-volume_roi_2)/volume_roi_1*100  
16.0672
```

This quantification gives an idea about the GM loss in percentage in these regions of interest. However, this longitudinal study is limited to two time instants only. It would be worth comparing the localization (D12) and quantification (degeneration_roi) of the degeneration at additional time instants, as well as statistically comparing it to a group of healthy subjects.

11.3 Going further

There is a large amount of different types of information available on medical image modalities (see the Introduction in this Chapter). All these in-

formation types can be efficiently combined by medical *image registration* (or *matching*). Image registration, which has not been handled in this Chapter, consists on finding the transformation that brings two medical images into a voxel-to-voxel correspondence. Many variables participate in the registration paradigm, making the classification of registration techniques a difficult task. A wide overview of medical image registration is done in (Maintz J.B.A. and Viergever M.A. 1998).

In image registration, two images are matched one to the other: the target image, also called *reference image* or *scene* (f), and the image that will be transformed, also called *floating image* or *deformable model* (g). If the images to register belong to the same modality it is said to be *monomodal* registration and if they belong to different modalities it is said to be *multi-modal* registration. We can also distinguish between *intra-subject* and *inter-subject* registration. In the first case, both reference and floating images belong to the same patient. The goal of intra-subject registration is usually to compensate the variable positioning of the patient in the acquisition process as well as other possible geometrical artifacts in the images. This kind of registration is usually needed in surgical planning, lesion localization or pathology evolution. In the second case, images are acquired from different patients. The goal of this kind of registration is to compensate the inter-subject anatomical variability in order to perform statistical studies or to profit from reference data¹³, for instance for surgical planning.

The image modalities to register, as well as the envisaged application, determine the remaining characteristics of the registration process:

- the nature and domain of possible transformations τ ,
- the features to be matched,
- the cost function to optimize.

The image registration problem can be formulated by the following minimization equation:

$$T^* = \underset{T \in \tau}{\operatorname{argmin}} \operatorname{cost}(f, T \circ g) \quad (11.31)$$

Note that all transformations in τ have to follow some physical constraints in order to model a *realistic* deformation between two brains (we can deform a brain into an apple but it is not very likely to happen).

¹³ Reference data is also called *Atlas* images that represent the human anatomy.

11.3.1 Nature and Domain of the transformation

In image registration, transformations are commonly either *rigid* (only translations and rotations are allowed), *affine* (parallel lines are mapped onto parallel lines), *projective* (lines are mapped onto lines) or *curved* (lines are mapped onto curves). Then, the transformation domain can either be *local* or *global*. A transformation is called global when a change in any one of the transformation parameters influences the transformation of the image as a whole. In a local transformation a change in any transformation parameter only affects a part of the image.

11.3.2 Features and cost function

The feature selection depends on the image modality and the definition of the cost function depends on the selected features.

The intensity image can be directly used as features for the matching process (this approach is called *voxel-based registration*) but other identifiable anatomical elements can be used such as point landmarks, lines or surfaces, which are previously extracted in both reference and floating images (this is called *model-based registration*).

Once feature selection is done, the cost function can be determined, commonly using:

$$\text{Cost function} = - \text{Similarity measure}.$$

The similarity measure can be intensity-based (in voxel-based registration) or distance-based (in model-based registration). Some typical similarity measures based on voxel intensity (Bankman I.N. 2000) are absolute or squared intensity difference, normalized cross-correlation, measures based on the optical flow concept¹⁴, information theory measures such as mutual information, etc. Some common distance-based measures are: Procrustean metric, Euclidean distance, curvature, etc.

11.3.3 Optimization

An optimization algorithm (Fig. 11.22) is needed to either maximize a similarity measure or minimize an error measure, over the search space defined by the parameters of the transformation. Various optimization strategies are available, such as Powell's or Simplex optimization, Steepest

¹⁴ The optical flow concept has been presented in Chapter 9 within the context of motion estimation.

Gradient or Genetic Algorithms. In fact, the strategy selection depends on the selected similarity measure since it may not always be possible to compute the derivatives with respect to the transformation parameters.

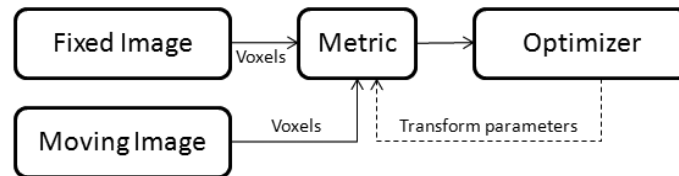


Fig. 11.22 Image Registration scheme

11.4 Conclusions

In this Chapter we have seen how GMMs and HMRFs are used for classifying images into homogeneous regions. GMMs are used to model the image intensity response and HMRFs are used to increase the model performance by including the spatial interactions across image voxels.

The brain tissue segmentation problem has been addressed, specially the detection and quantification of GM loss. The goal of the proof-of-concept Section was to perform a longitudinal study of 2 MR images. To this end, we have studied the image histogram and its fitting using a mixture of 3 Gaussian distributions. After that, we have applied the HMRF model and used the resulting probability maps to localize and quantify GM loss.

Finally, let us highlight that studies on patterns of brain atrophy in medical images are usually transversal studies; that is, they are performed statistically, in comparison to a probabilistic pattern that represents the normal anatomy. Nevertheless, the statistical classification framework that has been presented in this Chapter remains valid. It is important to note that registration, which has been briefly commented in Section 11.3, is needed in both longitudinal and transversal studies.

11.5 Acknowledgements

The authors would like to thank Prof. R. Meuli from the Radiology Department of the Lausanne University Hospital (CHUV) for providing the Magnetic Resonance images. This work is supported by the Center for Biomedical Imaging (CIBM) of the Geneva-Lausanne Universities, the EPFL, and the foundations Leenaards and Louis-Jeantet.

11.6 References

- Ashburner, J. and K. J. Friston (2000). "Voxel-based morphometry--the methods." *Neuroimage* 11(6 Pt 1): 805-21.
- Bach Cuadra, M., L. Cammoun, et al. (2005). "Comparison and validation of tissue modelization and statistical classification methods in T1-weighted MR brain images." *IEEE Trans Med Imaging* 24(12): 1548-65.
- Bankman I.N. (2000). *Handbook of Medical Imaging*. San Diego, Academic Press.
- Besag J. (1974). "Spatial interaction and the statistical analysis of lattice systems." *Journal of Royal Statistical Society* 36: 192-225.
- Bilmes, J. A. (1998). *A gentle tutorial of the EM algorithm and its applications to parameter estimation for Gaussian mixture and hidden Markov models*. Berkeley, ICSI.
- Galton, C. J., K. Patterson, et al. (2001). "Differing patterns of temporal atrophy in Alzheimer's disease and semantic dementia." *Neurology* 57(2): 216-25.
- Geman, S. and D. Geman (1984). "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images." *IEEE Trans. Pattern Analysis and Machine Intelligence* 6: 721-741.
- Geman S. and Geman D. (1984). "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images." *IEEE Trans. Pattern Analysis and Machine Intelligence* 6: 721-741.
- Gudbjartsson, H. and S. Patz (1995). "The Rician distribution of noisy MRI data." *Magn Reson Med* 34(6): 910-4.
- Haacke, E. M., R. W. Brown, et al. (1999). *Magnetic Resonance Imaging: Physical Principles and Sequence Design*, Wiley-Liss.
- Hajnal, J. V., D. L. G. Hill, et al. (2001). *Medical Image Registration*, CRC.
- Li, S. Z. (2001). *Markov Random Field Modeling in Image Analysis*, Springer.
- Maintz J.B.A. and Viergever M.A. (1998). "A survey of medical image registration." *Medical Image Analysis* 2(1): 1-36.
- Mazziotta, J. C., A. W. Toga, et al. (2000). *Brain mapping: The disorders*. San Diego, Academic Press.
- Suri, J. S., D. Wilson, et al. (2005). *Handbook of Biomedical Image Analysis: Segmentation Models (Volume 1 & 2)*, Springer.
- Van Leemput, K., F. Maes, et al. (2003). "A unifying framework for partial volume segmentation of brain MR images." *IEEE Trans Med Imaging* 22(1): 105-19.
- Zhang, Y., M. Brady, et al. (2001). "Segmentation of brain MR images through a Hidden Markov Random Field Model and the Expectation-Maximization algorithm." *IEEE Trans Med Imaging* 20(1): 45-57.