LILLE - BREST - TOULON
isen
ECOLE D'INGENIEURS

Institut Suprieur de l'Electronique et du Numrique
Tel. : +33 (0)2.98.03.84.00
Fax : +33 (0)2.98.03.84.10
CS 42807 - 29228 BREST Cedex 2 - FRANCE

N5
School Year 2008/2009
Training Period

Parameter space exploration and tools for fast visualization in EM segmentation and MRI bias field correction in Slicer 3

From April 6th to August 31th
At
Surgical Planning Laboratory (SPL)

Brigham & Women's Hospital
Harvard Medical School
75 Francis St.
Boston, MA 02115

Supervisors: Ron KIKINIS - SPL, Harvard Medical School - kikinis@bwh.harvard.edu
Sylvain JAUME - CSAIL, Massachusset's Insitut of Technologies - sylvain@csail.mit.edu

Referring Tearchers: Christine CAVARO-MENARD - M2 SIBM - christine.menard@univ-angers.fr
Dominique MARATRAY - ISEN Brest - dominique.maratray@isen.fr

by Nicolas RANNOU

Abstract

A couple of sentences on three or four lines to summarise your work.
This is a LaTeX template for undergraduate project reports.
Its detailed contents evolve to reflect FAQs.
Expectation-maximization is very popular for segmentation but it can be tricky to understand and to use. A full description of the EMS algorithm is done in this report. Different methods for fast parameters exploration are described. As part of the research, preprocessing methods like MRI bias field correction will be explained. The results obtained will be presented. Following the new workflow should allow the user to segment more datasets, more accurately.
Keywords: segmentation, expectation, maximization, correction, bias.

Resumé

Quelques phrases pour resumer mon travail.
C'est un template LaTeX pour les rapport.
Le contenu peut evoluer.
Mots clés :segmentation, expectation, maximisation, correction, biais.

Contents

List of Figures

Chapter 1

Introduction

1.1 Context and motivation

Nowadays, medical image processing is becoming a major field of research in most of the laboratories. Indeed, because of the increasing complexity of the data they have to deal with, physicists need something to help. Help must be provided in many different ways. Before the surgery, to etablish a fast and accurate diagnosis. During the surgery to prevent the physicist from errors and to help him to proceed to more precise moves. After the surgery, to see if the surgery succeed, or to follow the pathology of a patent. Nevertheless, the informations bring to the physicist by the tool must be accurate, robust and provide a fast feedback.

In this context of pre and post operation, plenty of work as already been done. Thus, there is still a lot of work to achieve. Regarding data storage and exchange, the increasing among of informations leads us to find other and more approriate methods. Another interesting contribution is images segmentation. New methods have to be developped for a better diagnosis, or to detect new pathologies. A lot of methods appears like level-set segmentation, region growing or texture based segmentation. Each one is adapted for a specific problem like vessels segmentation, tumors detection or SDFSF lungs detection. Another remarquable contribution is the segmentation based on expectation maximization (EM), which is very well suited for brain segmentation.

For the MR images segmentation purpose, the Surgical Planning Laboratory (SPL), Harvard Medical School and Brigham and Women's Hospital, has developped an EM algorithm to segment brain's MR images. The results obtained are very good until we segment small structures and we select the optimum parameters. The approach used for the intensity inhomogeinities estimation appears less efficient for the particuliar purpose of large structures segmentation WHY. Moreover, the implementation is not widely used so far, regarding the complexity of the segmentation process. In this report we will present an approach to enhance the segmentation for large structures, correctiong the intensity inhomogeinities in large structures and providing the end-user tools for an easier segmentation process.

1.2 Contents

The main body of this report is divided as follows.

Chap. 2 deals with the EM segmentation. Fundamentals will be reminded and the algorithm used will be described. We will also present in this chapter the limitations of the current implementation. Chap. 3 describes our contribution. It explains the solution we choose to enhanceme the segmentation and to improve the usability of the current framework. Chap. 4 shows the results achieved. It aslo discussed about what have been done, the limitations of the current module and the next work wich has to be done.

Chapter 2

Expectation-maximization applied to brain segmentation

Here we get going with theory of the expectation-maximization (EM) applied to brain segmentation. We show firstly a simple approach of the problem, in the particular case of Gaussian Mixture Models (GMM) followed by a more generalistic approach. The simple approach will give you an intuitive understanding of the problem then the general approach will formalize it in order to adapt it to most of the segmentation problems. Finally, there will be a presentation of the algorithm used in Slicer 3[1].

2.1 Presentation of the EM segmentation

The EM algorithm was originally described in 1977 by Arthur Dempster, Nan Laird, and Donald Rubin[1]. They generalized and developped a method used in several times by authors, for particular applications. It is widely used to solve problems where data are "missing". The EM algorithm is an iterative algorithm which works in two steps: Expectation and Maximization. It can be use to solve a lot of image processing's problems like classification, restoration[3], motion estimation[2], etc.. Since the generalization of the algorithm, a lot of related papers were proposed. Most of them bring algorithms derived from the original one to adapt it to particuliar problems using additional informations like spatial or structural information.

Nowadays, EM algorithms are become a popular tool for classification problems. It is particulary well suited for brain MR images segmentation. A lot of algorithms already exist. They present complex frameworks using spatial information, neighborhood or intensity inhomogeneities to enhance the classification.
In the SPL, the algorithm developped uses spatial, structural and intensity inhomogeneities informations to segment the brain.

2.2 Fundamentals

Here we get going with a presentation of all the fundamentals you need to have a good understanding of EM segmentation. We begin with a description of the statistical model used for the brain. Then we present briefly the widely used GMM. Finally we will introduce you to what the mamixum likelihood function. This part is mainly inspired from [4], [5] and [6]

2.2.1 Statistical model used for the brain

We define the voxel intensities of a MR image as $Y = \{y_1, ..., y_n\}$ when the image consisted in n voxels. Each y intensity is called *observed data* because this is the the data we see when we

[1] open source software developped in the SPL for biomedical engineering purpose

observe the image. Each y is a realization of the random variable Y. The real labelling of the image is Z. Z is called *hidden data* because we don't know the value of each label. This is precisely the purpose of the segmentation: estimating the *hidden data* from the *observed data*. We assume that the *observed data* is generated from the *hidden data* and a parameter Φ. The parameter Φ can either be a probability density function, noise, bias field, etc., depending on the model.

Y and Z can be viewed as n-dimensional random variables $Y = \{Y_1, ..., Y_n\}$ and $Z = \{Z_1, ..., Z_n\}$ then each y_i is a realisation of Y_i and each z_i is a realization of Z_i.The conditional probability function describing Y_i is $p(Y_i|Z_i, \Phi)$.

The easiest model assumes that each intensity in one class is the same, but this intensity is corrupted by factors like noise, with a Gaussian Distribution. We can describe the relationship as below:

$$y_i = \mu_k + n_i$$

where μ_k is the mean intensity of the k^th tissue and n_i a random sample generated by the corrupting factor(s). Let's say that n_i is generated by a Gaussian probability distribution function $G(., 0, \sigma)$, with 0 mean and σ variance. That means that y_i is a random sample generated by Gaussian probability density function $G(., \mu_k, \sigma)$. Let's assume that each class has a different variance, $G(., \mu_k, \sigma)$ becomes $G(., \mu_k, \sigma_k)$ and it leads to:

$$p(Y_i = y_i|Z_i = k, \Phi) = G(y_i, \mu_k, \sigma_k) \tag{2.1}$$

As the labelling is not known, it is usefull to express the probability density function (PDF) of Y_i only depending on parameter Φ with the total probability theorem:

$$p(Y_i|\Phi) = \sum_{k=1}^{K} p(Y_i = y_i|Z_i = k, \Phi)p(Z_i = k|\Phi) \tag{2.2}$$

$p(Z_i = k|\Phi)$ is the *prior probability*. It expresses the probability that a voxel i belongs to a class k. $p(Y_i = y|Z_i = k, \Phi)$ is the *likelihood*. In our case, we will assume that the *prior probability* is constant. The new model we obtain for the labelling, via a gaussian distribution, is a widely used one: the *Gaussian mixture model*.

2.2.2 Gaussian mixture model

Let's remind the first hypotesis: the conditional probability function for each tissue to segment is Gaussian equation (2.1). Moreover, we will assume that *prior probability* is a constant c_k for each class k. c_k is the *weight* of the class k.

$$p(Z_i = k|\Phi) = c_k \tag{2.3}$$

The last assumption will be that Φ contains unknown means, variances and weights for each tissue. Then we can express Φ as $\Phi = (\mu_1, \sigma_1, c_1, ..., \mu_K, \sigma_K, c_K)$.

Using equations (2.1) and (2.3), equation (2.2) becomes:

$$p(Y_i = y|\Phi) = \sum_{k=1}^{K} G(y, \mu_k, \sigma_k)c_k \tag{2.4}$$

In the case of *Gaussian mixture model*, each voxel is considered to be independent. That means that each voxel will have his own probability density function. Consequently, the normalized histogram of the whole volume can be interpreted as an approxmation of the sum of all the probability density functions.

2.2.3 Maximum likelihood

In our case, we know the intensity of each observed pixel y_i. Φ has to be found. The best estimation of Φ will be obtain using the maximum likelihood principle. $p(Y = y_i|\Phi)$ is called likellihood function and for it returns the value of the likelihood for y_i, given Φ. We can generalize it to the whole image with $p(Y|\Phi)$.

The voxels are considered to be independent through the whole volume. It leads us to:

$$p(Y|\Phi) = \prod_{i=1}^{n} p(Y_i|\Phi) \tag{2.5}$$

We must keep in mind that the objective is to find the parameter Φ which will maximize the likelihood of the observed volume. We can note this parameter:

$$\hat{\Phi} = \arg\max_{\Phi} p(Y|\Phi) \tag{2.6}$$

Therefore, it is more convenient to work with logarithm because the product from equation (2.5) will be converted into a sum. Equation (2.6) becomes:

$$\hat{\Phi} = \arg\max_{\Phi} \log p(Y|\Phi).$$

Let us denote:

$$\begin{aligned} L(\Phi) &\triangleq \log p(Y|\Phi) \\ &= \sum_{i=1}^{n} \log \sum_{k=1}^{K} p(Y_i = y|Z_i = k, \Phi) p(Z_i = k|\Phi) \end{aligned}$$

Finally, in case of GMM, with eq. (2.4), $L(\Phi)$ becomes:

$$L(\Phi) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} G(y, \mu_k, \sigma_k) c_k$$

The maximized *log* likelihood can then be computed using partial derivatives for each paraemter of Φ. When the partial derivative of $L(\Phi)$ is 0 for a parameter, we found the maximum likelihood for the parameter.

For example, to find the maximum likelihood for μ_k, we have to find when:

$$\frac{\partial}{\partial \mu_k}(L(\Phi)) = 0$$

Then we compute the partial derivative of $L(\Phi)$ over μ_k:

$$\begin{aligned} \frac{\partial}{\partial \mu_k}(L(\Phi)) &= \frac{\partial}{\partial \mu_k}(\sum_{i=1}^{n} \log \sum_{k=1}^{K} G(y, \mu_k, \sigma_k) c_k) \\ &= \sum_{i=1}^{n} \frac{G(y, \mu_k, \sigma_k) c_k}{\sum_{j=1}^{K} G(y, \mu_j, \sigma_j) c_j} \frac{\partial}{\partial \mu_k}(-\frac{(y - \mu_k)^2}{2\sigma_k^2}) \\ &= \sum_{i=1}^{n} \frac{G(y, \mu_k, \sigma_k) c_k}{\sum_{j=1}^{K} G(y, \mu_j, \sigma_j) c_j} (\frac{(y - \mu_k)}{\sigma_k^2}) \\ &= \sum_{i=1}^{n} \frac{p(Y_i = y|Z_i = k, \Phi) p(Z_i = k|\Phi)}{\sum_{j=1}^{K} p(Y_i = y|Z_i = j, \Phi) p(Z_i = j|\Phi)} (\frac{(y - \mu_k)}{\sigma_k^2}) \end{aligned} \tag{2.7}$$

Using Bayes' theorem (see App. A, Sec. A.2), we notice that:

$$p(Z_i = k|Y_i = y_i, \Phi) = \frac{p(Y_i = y_i|Z_i = k, \Phi)p(Z_i = k|\Phi)}{\sum_{j=1} p(Y_i = y_i|Z_i = j, \Phi)p(Z_i = j|\Phi)} \tag{2.8}$$

Thus, setting the denominator to 0 in equation (2.7) and using equation (2.8)yields:

$$\sum_{i=1}^{n} p(Z_i = k|Y_i = y_i, \Phi)(y - \mu_k) = 0 \tag{2.9}$$

Let us denote

$$p_{ik} = p(Z_i = k|Y_i = y_i, \Phi) \tag{2.10}$$

Equation (2.9) leads us to:

$$\mu_k = \frac{\sum_{i=1}^{n} y_i p_{ik}}{\sum_{i=1}^{n} p_{ik}} \tag{2.11}$$

Proceeding the same way as we did for equation (2.9), we can get similar equations for variance σ_j and weight c_j.

We find that:

$$\sigma_k^2 = \frac{\sum_{i=1}^{n} (y_i - \mu_k)^2 p_{ik}}{\sum_{i=1}^{n} p_{ik}} \tag{2.12}$$

$$c_k = \frac{1}{n} \sum_{i=1}^{n} p_{ik} \tag{2.13}$$

Equations (2.11),(2.12), and (2.13) provides us an equation for soft segmentation. This equation expresses that a voxel i belongs to the class k.

$$p_{ik} = \frac{G(y_i, \mu_k, \sigma_k)c_k}{\sum_{j=1}^{K} G(y_i, \mu_j, \sigma_j)c_j} \tag{2.14}$$

In the case og GMM, the segmentation can now be done following an iterative algorithm called *expectation maximization algorithm.*

2.3 Expectation maximization algorithm

The EM algorithm is a method to find the maximum likelihood for a given set of parameter (Φ in our case). Here we first get going with an intuitive description of the algorithm in the particular case of GMM then we will present a more general definition.

Algorithm in case of Gaussian mixture data model

Let's assume that we can find the maximum likelihood of the hidden data by a direct differentiation (because of the GMM). The EM algorithm is an iterative process of two steps: the expectation step (E-Step) and the maximization step (M-Step). At each iteration, the maximum likelihood will be increased until convergence is reached.

- **E-step**
  In this step, we calculate an estimation of soft segmentation $p^{(m+1)}$ with equation (2.14) as below. We know all the variables needed for the calculation from the observed data and the current parameter estimate $\Phi^{(m)}$. Note that an initialization is necessary for the first

iteration.

$$p_{ik}^{(m+1)} = \frac{G(y_i, \mu_k^{(m)}, \sigma_k^{(m)}) c_k^{(m)}}{\sum_{j=1}^{K} G(y_i, \mu_j^{(m)}, \sigma_j^{(m)}) c_j^{(m)}}$$

- **M-step**
  In this step, we estimate the maximum likelihood for parameter $\Phi^{(m+1)}$. We do it with equations (2.11),(2.12), and (2.13) as below. We know all the variables needed for the calculation from the observed data and the current estimate $p^{(m+1)}$ of hidden data.

$$\mu_k^{(m+1)} = \frac{\sum_{i=1}^{n} y_i p_{ik}^{(m+1)}}{\sum_{i=1}^{n} p_{ik}^{(m+1)}}$$

$$(\sigma_k^{(m+1)})^2 = \frac{\sum_{i=1}^{n} (y_i - \mu_k^{(m+1)})^2 p_{ik}^{(m+1)}}{\sum_{i=1}^{n} p_{ik}^{(m+1)}}$$

$$c_k^{(m+1)} = \frac{1}{n} \sum_{i=1}^{n} p_{ik}^{(m+1)}$$

The problem is simple as long as we are working with GMM. In the other case, the log-likelihood can not be maximized by direct differenciation and a more generalized approach must be used.

Generalized algorithm

Now we assume that we are no longer working with GMM. Thus, we must use a more general algorithm. To explain the general algorithm, we will start from the log-likelihood $L(\Phi)$. As presented in the previous subsection:

$$L(\Phi) = \log p(Y|\Phi)$$

Since *log* is a strictely increasing function, the value of Φ which will maximizes $p(Y|\Phi)$ will also maximizes $L(\Phi)$. We want to maximize $L(\Phi)$. Thus, after the m^{th} iteration, we want an estimated Φ_n such as:

$$L(\Phi) > L(\Phi^{(m)})$$

In other words, we want to maximize the difference $L(\Phi) - L(\Phi_n)$. For convenience, we introduce a new variable z_{ik} which means that $Z_i = k$. Using the new notation, we can transform this difference as below:

$$
\begin{aligned}
L(\Phi) - L(\Phi_n) &= \log p(Y|\Phi) - \log p(Y|\Phi_n) \\
&= \sum_{i=1}^{n} \{\log \sum_{k=1}^{K} p(Y_i|z_{ik}, \Phi) p(z_{ik}|\Phi) - \log p(Y_i|\Phi_n)\} \\
&= \sum_{i=1}^{n} \{\log \sum_{k=1}^{K} p(Y_i|z_{ik}, \Phi) p(z_{ik}|\Phi) . \frac{p(z_{ik}|Y_i, \Phi_n)}{p(z_{ik}|Y_i, \Phi_n)} - \log p(Y_i|\Phi_n)\} \\
&= \sum_{i=1}^{n} \{\log \sum_{k=1}^{K} p(z_{ik}|Y_i, \Phi_n) . \frac{p(Y_i|z_{ik}, \Phi) p(z_{ik}|\Phi)}{p(z_{ik}|Y_i, \Phi_n)} - \log p(Y_i|\Phi_n)\} \\
&\geq \sum_{i=1}^{n} \{\sum_{k=1}^{K} p(z_{ik}|Y_i, \Phi_n) \log \frac{p(Y_i|z_{ik}, \Phi) p(z_{ik}|\Phi)}{p(z_{ik}|Y_i, \Phi_n)} - \log p(Y_i|\Phi_n)\}
\end{aligned}
$$

We can deduce this inequality from Jensen's inequality (see App. A, Sec. A.3) since $p(z_{ik}|Y_i, \Phi^{(m)})$ is a probability mesure and ln a concave function ([5]).

We will then use the fact that $\sum_k p(z_{ik}|Y_i, \Phi^{(m)}) = 1$. In this case, it leads to $\log p(Y|\Phi^{(m)}) = \sum_k p(z_{ik}|Y_i, \Phi^{(m)}) \log p(Y|\Phi^{(m)})$. This allows us to bring $\log p(Y|\Phi^{(m)})$ into the summation.

We will also use a new variable e_k to express the difference through the whole volume without a summation. e_k is defined as $e_k = \{z_{1k}, ..., z_{nk}\}$ when the image consisted in n voxels. For example, $z_{ik} = e_k = \{0, ..., 0, 1, 0, ..., 0\}$ means that voxel i belongs to class k.

$$
\begin{aligned}
L(\Phi) - L(\Phi_n) &\geq \sum_{i=1}^{n} \{\sum_{k=1}^{K} p(z_{ik}|Y_i, \Phi^{(m)}) \log \frac{p(Y_i|z_{ik}, \Phi) p(z_{ik}\Phi)}{p(z_{ik}|Y_i, \Phi^{(m)})} - \log p(Y|\Phi^{(m)})\} \\
&= \sum_{i=1}^{n} \{\sum_{k=1}^{K} p(z_{ik}|Y_i, \Phi^{(m)}) \log \frac{p(Y_i|z_{ik}, \Phi) p(z_{ik}|\Phi)}{p(z_{ik}|Y_i, \Phi^{(m)}) p(Y_i|\Phi^{(m)})}\} \\
&= \sum_{k=1}^{K} p(e_k|Y, \Phi^{(m)}) \log \frac{p(Y|e_k, \Phi) p(e_k|\Phi)}{p(e_k|Y, \Phi^{(m)}) p(Y|\Phi^{(m)})} \\
&\triangleq \Delta(\Phi|\Phi_n
\end{aligned}
$$

We can then conclude that:

$$
\begin{aligned}
L(\Phi) &\geq L(\Phi^{(m)}) + \Delta(\Phi|\Phi^{(m)}) \\
&\geq l(\Phi|\Phi^{(m)})
\end{aligned}
$$

where $l(\Phi|\Phi^{(m)}) \triangleq L(\Phi(m)) + \Delta(\Phi|\Phi^{(m)})$.

We have now a function $l(\Phi|\Phi^{(m)})$ which is bounded above by $L(\Phi)$. The only time when the two functions are egals is when $\Phi = \Phi^{(m)}$. Our objective is to find the value of Φ which will maximize $L(\Phi)$.
We can formalize this research. $\Phi^{(m+1)}$ is the updated value which is get after maximization of Φ using $\Phi^{(m)}$.

$$
\begin{aligned}
\Phi^{(m+1)} &= \arg\max_{\Phi}\{l(\Phi|\Phi^{(m)})\} \\
&= \arg\max_{\Phi}\{L(\Phi^{(m)}) + \sum_{k=1}^{K} p(e_k|Y,\Phi^{(m)}) \log \frac{p(Y|e_k,\Phi)p(e_k|\Phi)}{p(e_k|Y,\Phi^{(m)})p(Y|\Phi^{(m)})}\} \\
&\text{As } p(e_k|Y,\Phi^{(m)}) \text{ and } p(Y|\Phi^{(m)}) \text{ do not depend on } \Phi \\
&= \arg\max_{\Phi}\{\sum_{k=1}^{K} p(e_k|Y_i,\Phi^{(m)}) \log p(Y|e_k,\Phi)p(e_k|\Phi)\} \\
&= \arg\max_{\Phi}\{\sum_{k=1}^{K} p(e_k|Y,\Phi^{(m)}) \log \frac{p(Y,e_k,\Phi)p(e_k,\Phi)}{p(e_k,\Phi)p(\Phi)}\} \\
&= \arg\max_{\Phi}\{\sum_{k=1}^{K} p(e_k|Y,\Phi^{(m)}) \log p(Y,e_k|\Phi)\} \qquad (2.15) \\
&= \arg\max_{\Phi}\{E_{Z|Y,\Phi^{(m)}}\{\log p(Y,Z|\Phi)\}\}
\end{aligned}
$$

We notice from equation (2.15), that , for a given voxel i:

$$\sum_{k=1}^{K} p(e_k|Y_i,\Phi^{(m)}) \log p(Y_i,e_k|\Phi) = \sum_{k=1}^{K} p_{ik}^{(m+1)} \log p(Y_i,e_k|\Phi) \qquad (2.16)$$

Now, both expectation and maximization steps are apparents.

- **E-step**
  This this the expectation step. During this step, we estimate the probability that the pixel i belongs to class k regarding $\Phi_{(m)}$. This equation is obtained from equation (2.16) with Bayes formula.

$$p_{ik}^{(m+1)} = \frac{p(Y_i|e_k,\Phi^{(m)})p(e_k|\Phi^{(m)})}{\sum_{j=1}^{K} p(Y_i|e_j,\Phi^{(m)})p(e_j|\Phi^{(m)})} \qquad (2.17)$$

  p_{ik} is used to fill a "map" of soft segmentation. At the end of the segmentation, This map contains the probability that the voxel i belongs to class $1, 2, ..., K$. We determine the class of the pixel i looking at the class which has the highest probability for this given voxel i. Using this probability, $E_{Z|Y,\Phi^{(m)}}$ returns the expected value of the parameter Φ regarding $\Phi^{(m)}$.

- **M-step**
  This this the maximization step. During this step $\arg\max_{\Phi}$ maximizes $E_{Z|Y,\Phi^{(m)}}$ for the parameter Φ. It returns $\Phi^{(m+1)}$

$$\arg\max_{\Phi}\{E_{Z|Y,\Phi^{(m)}}\{\log p(Y,Z|\Phi)\}\} \qquad (2.18)$$

The EM algorithm iterates until convergence is reached. The condition of convergence can differ from an algorithm to another. A possibility is to fix the number of iterations of the algorithm. Most of the time, the following approach is used; convergence is reached when the difference between the estimation of the parameter Φ, at step m and step $m+1$ is smaller than ε:

$$\Phi^{(m+1)} - \Phi^{(m)} < \varepsilon$$

If after the C^{th} iteration, this condition is not satisfied, the EM algorithm is stopped.

We are now familiar with the theory behind the EM algorithm. The logic appears clearly and we can summarize the basic algorithm with the figure (2.1):

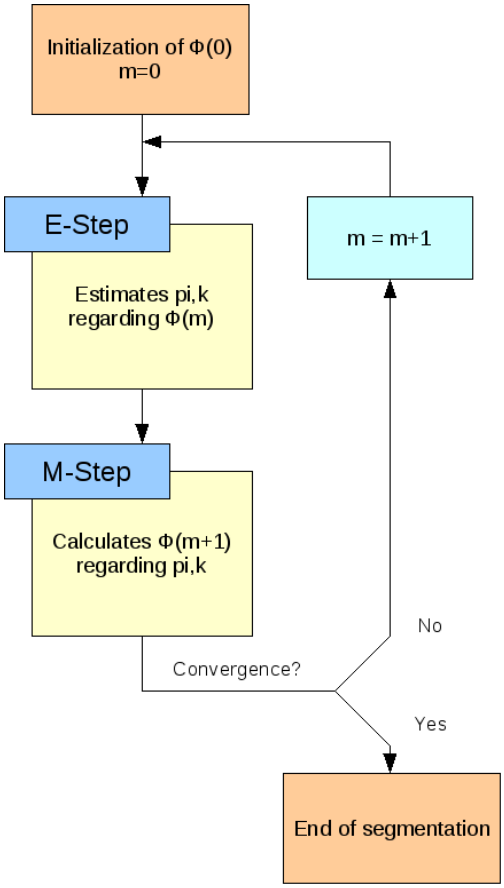

Figure 2.1: Basic EM algorithm

2.4 Expectation maximization algorithm used in Slicer 3

In this part, we present the EM algorithm which has been intergated in Slicer 3 and the worklow in which it ahs been integrated as well. Finally, we briefly discuss of the limitations of this one.

The EM algorithm which is used in the SPL is a derived one, from the original one. It enhances the original algorithm adding informations like a probalistic atlas, a multichannel segmentation, a bias correction and a structure information. In Slicer 3, the GMM is used to describe the tissues to segment. Thus it will simplify the problem and the notations.

2.4.1 Probabilistic atlas

The EM algorithm is very sensitive to initialization since it only finds local extremums during the maximization step. A solution to enhance the initialization is to use atlases. There is an atlas for each class you want to segment. For each voxel i of the volume, it returns the probability that this voxel belongs to class k. This probability can be used as initialization.

$$p_{ik}^{(0)} = p_{ik}^{atlas}$$

From this value, it estimates $\Phi^{(1)}$ and the algorithm iterates until convergence is reached. The probabilistic atlases are not only used to inialize the process. It is also use to get a more robust algorithm. Indeed, we can use the spatial information given by the atlases. Voxels will be classified not only based on intensity but regarding spatial position as well. Van Lemput *et al.* ([8] and [9]) used the spatial prior at each iteration. It is constant and we then have a spatial information. The probability that a pixel i belongs to class k, in the E-Step changes. From equation (2.17), it becomes:

$$p_{ik}^{(m+1)} = \frac{p(Y_i|e_k, \Phi^{(m)} p_{ik}^{atlas}}{\sum_{j=1}^{K} p(Y_i|e_j, \Phi^{(m)}) p_{ik}^{atlas}}$$

2.4.2 Multichannel segmentation

Most of the time, several modalities are used to process brain segmentation. Indeed, the best suited modality to use depends on the tissue you want to segment. For example, T1[2] MR images are well suited to segment white matter (WM) but are really bad for cerebrospnial fluid (CSF). On the contrary, T2[3] MR images are well suited for CSF and not for WM. To formalize the utilisation of different MR images sequences during the segmentation, we will change the definition of y_i and of μ_k we did at the beginning.Let $y_i = \{y_{i1}, y_{i2}, ..., y_{iR}\}$ and $\mu_i = \{\mu_{i1}, \mu_{i2}, ..., \mu_{iR}\}$ when we use R images, from different modalities to do the segmentation. The equations for the E-Step and the M-Step will remain the same.

2.4.3 Bias field correction

A major issue in MR modality is that the images can be corrupted by a low field bias field. It is due to equipment limitations or/and to patient induced electrodynimic interactions ([12]). We will now present how this bias field can be estimated and corrected in the EM algorithm.

Principle

Let $I = (I_1, ..., I_n)$ the observed intensities in an image, $I^* = (I_1^*, ..., I_n^*)$ the ideal intensities and $F = (F_1, ..., F_n)$ the bias field. Then, the degradation at each voxel can be expressed as:

$$I_i = I_i^* F_i$$

Let $Y = (Y_1, ..., Y_n)$ and $Y^* = (Y_1^*, ..., Y_n^*)$ be the log-transformed observed and ideal intensities.$B = (B_1, ..., B_n)$ the log-bias field. This transforms makes the bias field becomes additive instead of multiplicative without the log-approach.

$$Y_i = Y_i^* + B_i$$

We can model the PDF of the voxel intensity with a gaussian distribution

$$p(y_i|e_k, \Phi, B) = G(y_i - b_i, \mu_k, \sigma_k)$$

The low frequency characteristic of the bias field B can be modeled by a linear combination of smooth basics functions $\Psi_l(x)$ ([13]). Let b_i be the realisation of the random variable B_i

$$b_i = \sum_{l=1}^{L} a_l \Psi_l(pos(i))$$

[2]MR images acquisition sequence designed to enhance the grey matter/white matter contrast. See [12].
[3]MR images acquisition sequence designed to enhance the grey matter/cerebrospinal fluid contrast. See [12].

$pos(i)$ returns the 3D position (x, y, z) of the voxel i. a_i is the $i^{(th)}$ value of the vector $A = (a_1, ..., a_L)$. A represents the bias field parameters.

In the GM model, bias field can then be estimated using EM framework. The bias field parameter A will be used during the E-Step , through bi to estimate the soft segmentation. A will be re-estimated during the M-Step, after the maximization of the tissue class parameters (mean, variance and weight). Van Leemput formalised the two steps as below ([8] and [9]):

- **E-step**

$$p_{ik}^{(m+1)} = \frac{G(y_i - b_i, \mu_k^{(m)}, \sigma_k^{(m)}) p_{ik}^{atlas}}{\sum_{j=1}^{K} G(y_i - b_i, \mu_j^{(m)}, \sigma_j^{(m)}) p_{ij}^{atlas}}$$

- **M-step**

  - Gaussian distribution parameters estimation

$$\mu_k^{(m+1)} = \frac{\sum_{i=1}^{n} y_i p_{ik}^{(m+1)} - b_i}{\sum_{i=1}^{n} p_{ik}^{(m+1)}}$$

$$(\sigma_k^{(m+1)})^2 = \frac{\sum_{i=1}^{n} (y_i - \mu_k^{(m+1)} - b_i)^2 p_{ik}^{(m+1)}}{\sum_{i=1}^{n} p_{ik}^{(m+1)}}$$

  - Bias field correction

$$(A^{(m+1)})^T = (F^T W^{(m+1)} F)^{-1} F^T W^{(m+1)} R^{(m+1)} \tag{2.19}$$

with:

$$\mathbf{F} = \begin{pmatrix} \Psi_1(pos(1)) & \Psi_2(pos(1)) & \dots & \Psi_L(pos(1)) \\ \Psi_1(pos(2)) & \Psi_2(pos(2)) & \dots & \Psi_L(pos(2)) \\ \vdots & \vdots & \ddots & \\ \Psi_1(pos(N)) & \Psi_2(pos(N)) & \dots & \Psi_L(pos(N)) \end{pmatrix}$$

$$\mathbf{W^{(m+1)}} = \begin{pmatrix} \sum_{k=1}^{K} w_{1k}^{(m+1)} & 0 & \dots & 0 \\ 0 & \sum_{k=1}^{K} w_{2k}^{(m+1)} & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & \dots & 0 & \sum_{k=1}^{K} w_{Nk}^{(m+1)} \end{pmatrix}$$

$$w_{ik}^{(m+1)} = \frac{p_{ik}^{(m+1)}}{(\sigma_k^{(m+1)})^2}$$

$$\mathbf{R} = \begin{pmatrix} y_1 - \tilde{y}_1^{(m+1)} \\ \vdots \\ y_N - \tilde{y}_N^{(m+1)} \end{pmatrix}$$

$$\tilde{y}_i^{(m+1)} = \frac{\sum_{k=1}^{K} w_{ik}^{(m+1)} \mu_{ik}^{(m+1)}}{\sum_{k=1}^{K} w_{ik}^{(m+1)}}$$

The bias field correction can be interpreted as follows: the estimated soft segmentation $(p_{ik}^{(m)})$ and tissue class parameters are used to reconstruct the image $\tilde{Y} = (\tilde{y_1}, ..., \tilde{y_n})$. This new image is supposed not to be corrupted by the bias field. We then substract the reconstructed image $\tilde{Y}$ from the observed image Y. We obtain the residual image R. From R, we estimate the bias field. F represents the discretized geometry of the bias field. W is an inverse covariance matrix. It returns informations about the possible error for each voxel. The covariance matrix will be descibed in details in section (QSDQSD).

The approach used in Slicer 3 is based on the same principle but differs regarding the maximization method and the parameter which is maximized.

Variation used in Slicer 3

In this method, we are working with a GMM. Moreover the parametres of this gaussian distribution are assumed to be known. The idea of estimating the field in EM framework was originally proposed by Wells *et al.* ([10]). He proposed to only use maximization to re-estimate the bias field. The *maximum a posteriori principle* (MAP) instead of the maximum likelihood principle (equation (2.6)) is used to find the lower bound, duringthe maaximization:

$$\begin{aligned}
\hat{\Phi} &= \arg\max_{\Phi} p(\Phi|Y) \\
&\text{As Bayes's theorem can be applied} \\
&= \arg\max_{\Phi} \frac{p(Y|\Phi)p(\Phi)}{pY} \\
&\text{As } p(Y) \text{ do not depend on } \Phi \\
&= \arg\max_{\Phi} p(Y|\Phi)p(\Phi)
\end{aligned}$$

Proceeding the same way as we did in section 2.3, the new E-Step becomes:

$$E_{MAP} = E_{Z|Y,\Phi^{(m)}}\{\ln p(Y, Z|\Phi)\} + \ln p(\Phi) \tag{2.20}$$

In Wells' method, the only parameter to be estimate is the bias field. We assume that the noise has a gaussian distribution:

$$p(\Phi) = p(B) = G(B, 0, \Sigma_B)$$

The equation for the bias field will change. We add the smoothness constraint in the gaussian distribution: Σ_B^{-1}. We also set F to unix matrix as no parametric model for the bias field is assumed. Finally, we define the mean residual image $\bar{M}^{(m+1)}$

$$\bar{M}^{(m+1)} = W^{(m+1)} R^{(m+1)}$$

The equation (2.19) for the bias field will then be remplaced by $(B^{(m+1)})^T$:

$$(B^{(m+1)})^T = (W^{(m+1)+\Sigma_B^{-1}})^{-1} \bar{M}^{(m+1)}$$

2.4.4 Hierarchical information

The last modification of the original EM algorithm is the addition of hierarchical information in the iterative segmentation process. The algorithm was described by Pohl *et al.* ([11]) The idea was to describe the structures we want to segment as a tree. It allows us to subdivise the segmentation process into subproblems, that are easier to solve, according to Pohl.

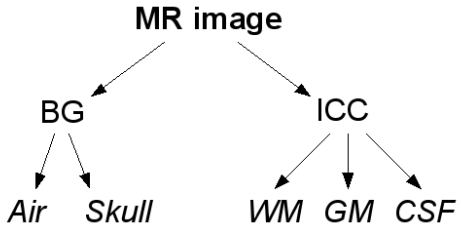

Figure 2.2: A simple tree structure of the brain

Here we continue with an intuitive description of the process. It is a brief explanation of how the tree figure (2.2) would be segmented using the tree structure information. At the first iteration, the MR image will be segmented into the background (BG) and the intracranial cavity (ICC) with the EM algorithm. At the second iteration, the BG will be segmented into the air and the skull. Finally, at the last iteration, the ICC will be segmented into white matter (WM), grey matter (GM) and cerebrospinal fluid (CSF).

To formalize it, we incorporate H, a set of structure-specific information in equation (2.20). H contains a lot of information like the structures of the tree which have to be segmented, an approximative size of the structure to be segmented and information about which modality is the best suited to segment this structure.

$$\Phi^{(m+1)} = \arg\max_{\Phi}\{\sum_{k=1}^{K}(p(e_k|Y,\Phi^{(m)},H)\log p(Y,e_k|\Phi,H)) + \ln p(\Phi,H)\}$$

2.4.5 Summary

We have shown that the EM algorithm is very flexible and can be transformed to solve a lot of segmentation problems. It is very well suited for segmentation of MR brain images and we can add a lot of informations through this algorithm to enhance the segmentation. The iterative general process is divided in two steps: the expectation (E-Step) and the maximization (the M-Step).

- **E-Step**
  Estimates a soft segmentation (p_{ik}), given parameter $\Phi^{(m)}$. The soft segmentation creates a map of probability. Each voxels contains the probabilities that it belongs to each class. It is used for the final segmentation.

- **M-Step**
  Estimates $\Phi^{(m+1)}$, using the soft segmentation done in the E-Step.

  - Estimates the intensity distribution for each intensity class.
  - Estimates the bias field, $\Phi^{(m+1)}$, using the soft segmentation and the intensity class distribution

Each node will then be segmented until the whole tree has been processed as described in section (2.4.4).

We can also describe the segmentation process in the figure (2.3). To describe how it works, we use the tree structure presented in figure (2.2) . It first segments the node $n = 0$, i.e., BG and ICC. Once the EM Segmentation has converged, BG and ICC have been segmented and we move to the next node. Air and Skull will then be segmented. Following this process, all the structures of the tree will be segmented.

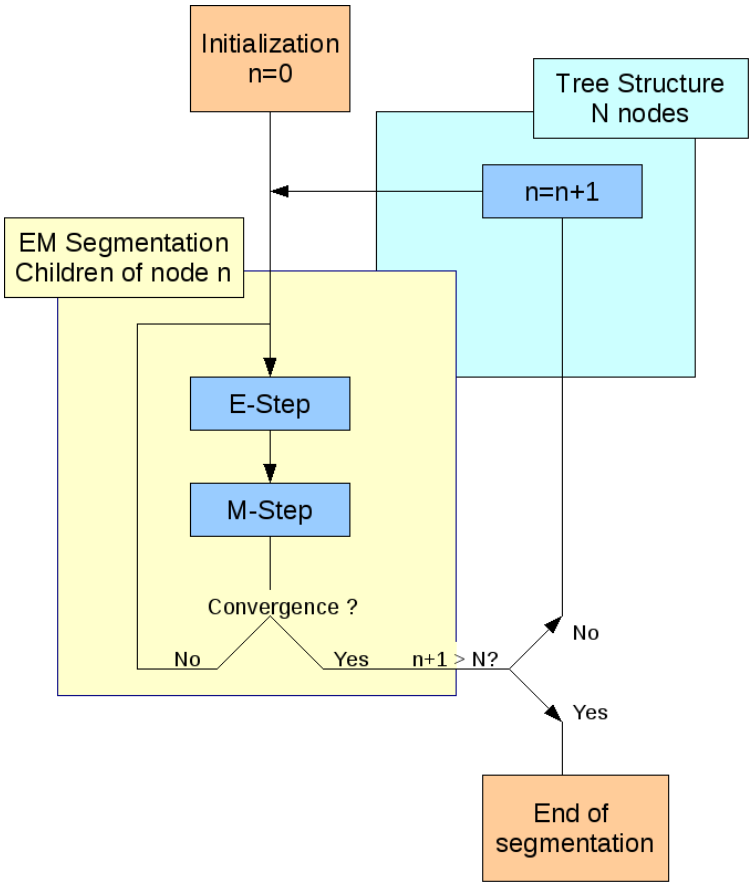

Figure 2.3: EM segment algorithm in Slicer

2.5 Workflow in Slicer 3

We will now present the whole segmentation framework used in Slicer 3. This will describe all the initialisation steps done by the user, via the graphical user interface (GUI). It will also present how the whole algorithm works. We will remind why is each initialisation step important and where is this information used.

2.5.1 User interface

It consists in a manual initialization of the parameters which are require for the segmentation on Slicer 3. The user chooses the good values via the GUI.

- **Step 1: Tree structure creation**
  We first create the tree structure we want to use for the segmentation. It will be used to define H (section (2.4.4)).

- **Step 2: Atlas assignment**
  We assign to each node of the tree, i.e. to each tissue to segment, the related atlas. It will be used for the spatial information (section (2.4.1)). It implies that you need an atlas for each structure that you want to segment.

- **Step 3: Multimodal segmentation**
  We choose the images we want to use for the segmentation. As discussed in section (2.4.2), it is usefull for the mutlichannel segmentation since some tissues are not clearly visible in some modalities.

- **Step 4: Intensity normalization**
  We choose the value for the intensity normalization. We normalize the intensity of the images to be segmented, regarding the atlas. The target images are normalized. Thus, they have the same mean intensity as the related atlas. It is usefull if we want to use the command-line module. Doing an intensity normalisation, the initialisation value of each class will be the same for each volume. It is a convenient way to run a lot of segmentation.

- **Step 5: Class definition**
  We define mean value, variance and covariance for each class and modality. Moreover, it is precise way to initialiase the class tissues distribution for the algorithm. It it usefull because the EM algorithm used only estimates the bias field but still requires informations about the tissues to be segmented. Using these values, the algorithm is can be initialized then estimates the bias.

- **Step 6: Prior weights**
  We define the importance of each target image and of the atlas for each tissue to segment. It is will be usefull to define H (section 2.4.4). For each tissue, H knows usefull informations like which modality is the more relevant for the class and the size that the class is supposed to be.

  For example, for the CSF. Let's assume that we proceed to a multi channel segmentation with T1 and T2 MR images. We give a weight of one (maximum) to the T2 target volume and zero (minimum) to the T1 target volume. It means that the only relevant information for CSF will be in the T2 volume. The algorithm will act in consequence and only use the information from T2 to do the segmentation. It is the same for the atlas. If we set the weight of the atlas to one, the algorithm will use the spatial information. If we set it to zero, it will not.

- **Step 7: Registration method**
  We choose the type of registration we want. Different kind of registrations are available. The default registration method is a non-rigid registration. The point of choising a registration method is presented in the next section.

2.5.2 Algorithm

After all the initialisation steps done via the GUI (section 2.5.1), we will no present the core of the segmentation pipeline.

- **Step 1: Intensity normalisation**
  The intensity of the target volumes are normalized to the value that the user chose in the GUI. The utility of this treatment has been discissed in the previous section, at the step 4.

- **Step 2: Images registration**
  In order for the atlas to guide the segmentation (spatial information), it as to be aligned to the target images. The transformation is evaluated between the first normalized target image and its atlas.

- **Step 3: Spatial prior alignement**
  The transformation computed during the image registration is applied to all the structure-specific atlases.

- **Step 4: EM Algorithm in the tree structure**
  The whole segment workflow is then applied (section 2.4.5).

2.5.3 Summary

In Slicer 3, the whole segmentation pipeline can be described as in figure (2.4). There is first an iniatialization step, done by user via the GUI. Then, some pre-processing steps are applied in order to enhance the segmentation. Finally, the EM algorithm segments the MR images.

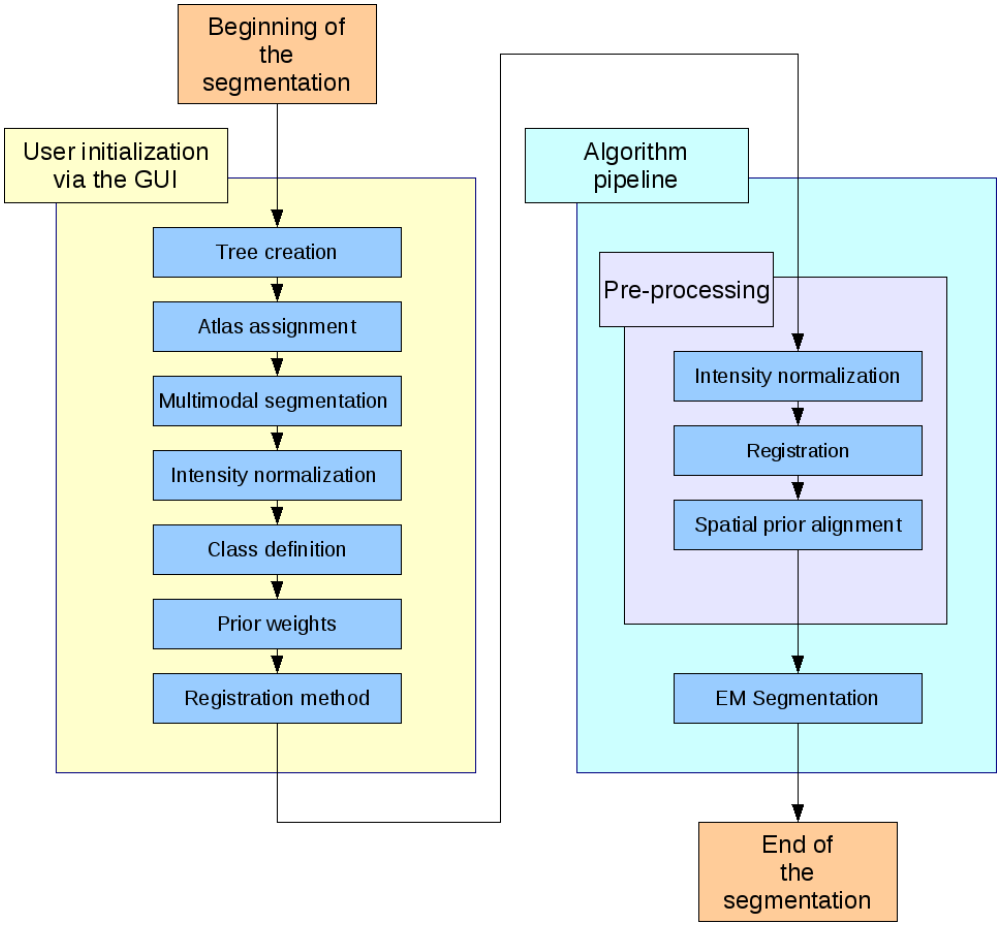

Figure 2.4: The whole segmentation pipeline in Slicer 3

2.6 Limitations

Even if the EM segmentation pipeline is robust in Slicer 3, some limitations appear.

The first problem the user has to face appears during the intensity normalization step. The user has no tools to find the good normalization value. He has to guess it. This problem will be discussed in section SDFSDF.

The second problem is directly linked to the EM algorithm. As the maximization method is a local one, the class distribution, which are used for the initialization of the algorithm have to be well defined. So far, we have no possibilities to know how accurate our definition of the class is. This problem will be discussed in details in section SDFSDF.

Another problem appears at the same steps. The actual method for defining means and variances for each class appears not as efficient as we want it to be. This problem will be discussed in details in section SDFSDF.

The last problem we encountered appeared after the initialization, during the pre-processings. Only one pre-processing (intensity normalization) is done on the target images before the registration to the atlas. Since the EM algorithm is mainly designed to work on MR images, bias field is a recurrent problem. Proceeding to a registration before a bias correction can deteriore a lot the

results of the registration. This problem will be discussed in details in section SDFSDF.

After this presentation of the main issues, we will in the next chapter explain the problems more deeply and show you the solutions we brought to solve it.

Chapter 3

The contributions

In this chapter we will present all the contribution to enhance the segmentation workflow in Slicer 3. We propose solution to the problem cited in the previous chapter (2.6). We get started with the registration problem. The we propose solutions to enhance the class selection and to allow the user to evaluate his selection. Finally, we present some tools we added to help the user to find the good intensity normalization value and an estimation of the global prior weights.

3.1 MRI Bias Field correction

The registration step could present some problems if the image to segment has intensities inhomogeinities. We remind the problem, then we present the solution proposed.

3.1.1 Interest

In the segmentation process, a registration step is required. Registration consists in finding a trasformation to fit two images as well as possible. It is described in details in DSFSDF. Only one pre-processing (intensity normalization) is done before the registration. The problem is that the algorithm is designed to treat MR images. MR images are often corrupted by a bias field. Thus, the the image to register presents intensities inhomogeinities. These inhomogeinities can deteriore a lot the registration.

On figure (3.1), we present the result of the registration between an atlas and a biased MR image. Note that the target MR image has been normalized to have the same mean value as the atlas. The results is clearly bad. A solution must be brought to enhance this step and so the segmentation.

3.1.2 Our approach

The idea simply consists in correcting the bias field of the MR image before this step. Thus, the registration will be significantly enhanced.Since the registration is better, it should also increase the segmentation.

To correct the bias field, we used the non-parametric approach presented by Styner in SDFSDF. We choose a non-parametric approach because it doesn't require prior information like the number of tissue to correct or the mean value of each tissue to correct. We implement an ITK[1] filter ([14]) in Slicer 3.

We choose not to implement it in Slicer 3 as part as the EM Segment module. Indeed, users may want to correct the bias field in MR images for other purposes. Moreover, because it would be the first pre-processing step, it is possible to do so. The user will first have to correct the

[1] open-source C++ toolkit for segmentation and registration. See [13].

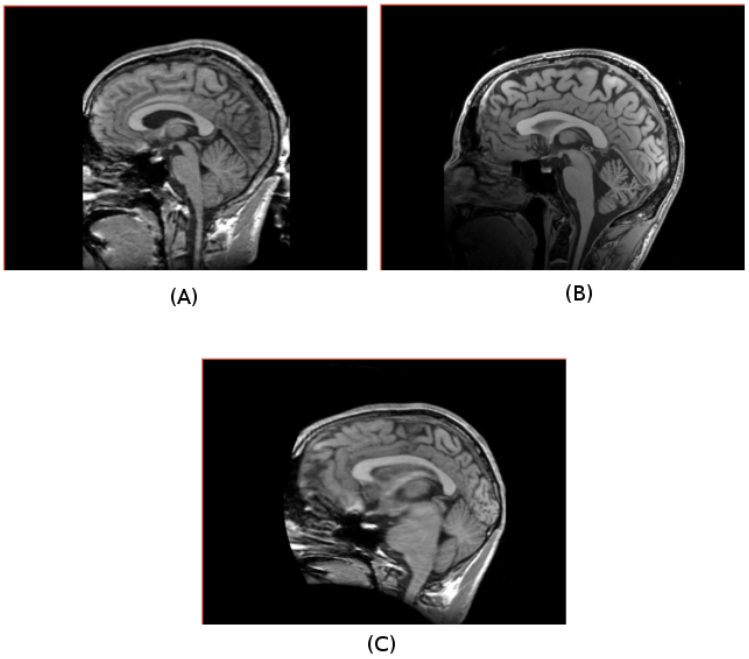

Figure 3.1: Result of registration of a biased MR image without correction

intensities inhomogeinities via the module then use the corrected images in the EM segmentation module.

We can describe the new segmentation workflow in Slicer 3 as we do in figure (3.2).

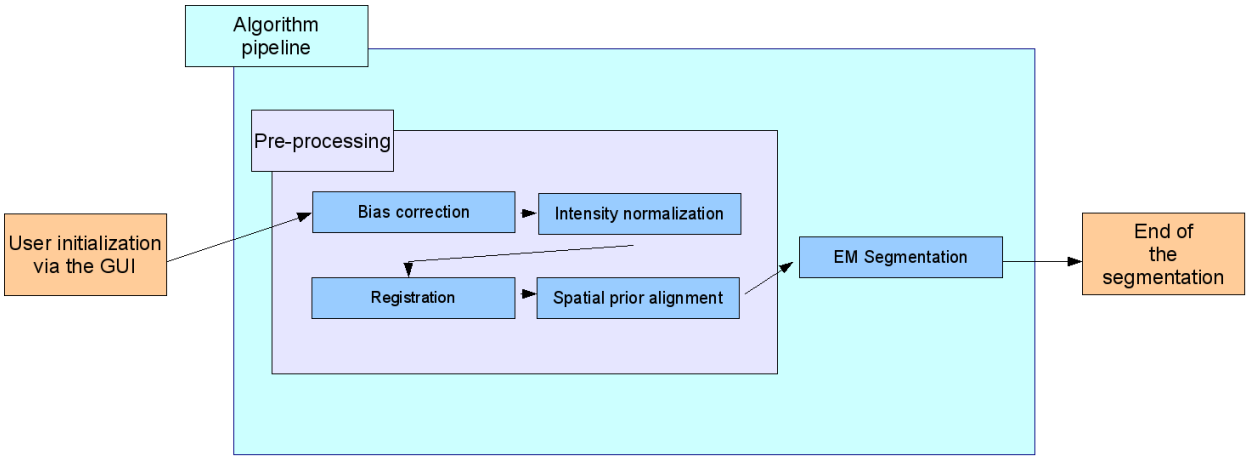

Figure 3.2: New algorithm pipeline

After the bias correction, the result of the registration clearly appears to be better (3.3).

We evaluated accuracy of the registration. We used the simple approach of the joint histogram. The joint histogram evalutaion method is basic comparaison. Let A be a matrix of size $W * L$. W will be the intensity range of the first image used for the comparaison. L will be the intensity range of the second image to be coompared. The matrix is initialized to 0. Each time that in the same position, there is the same intensities in the two images, we add 1 in the good cell in the matrix. After the joint histogram creation, the value at the coordinate i, j in the matrix is the number of pixel pairs having gray level i at position x, y.

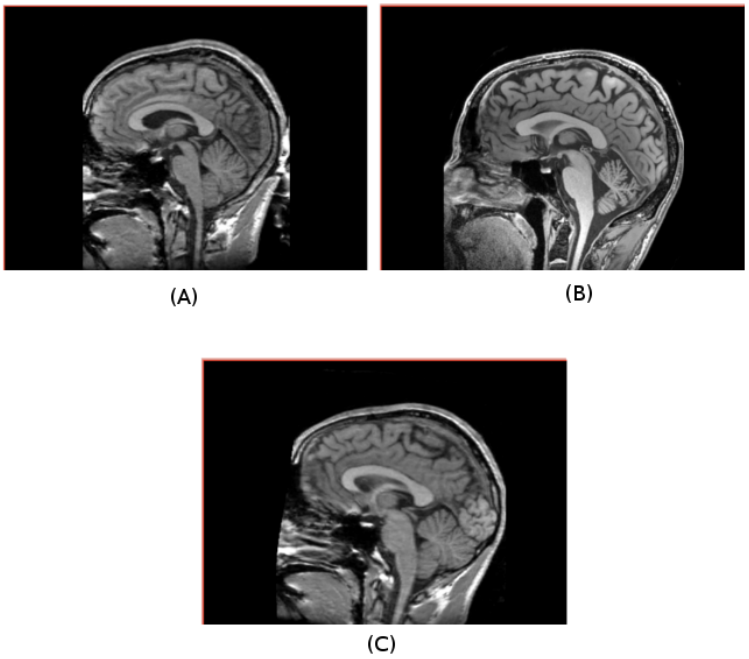

Figure 3.3: Registration after bias correction.

3.2 Class Distribution selection

During parameters initialization, the user has to define each class distribution. The previous method of selection presented some limitations and we proposed a new approach.

3.2.1 Interest

So far, the user had two choices to define each class distribution.

The first possiblity consisted in entering manually the intensities mean value and variance for each class, for each volume to be processed. This way, the user can be very precise and accurate when he defines each class. But it is very hard to find the good mean value and variance for each class for each volume. Morever, each time we want to process a new volume, we have to redefine mean values and variances. It is not convenient and it can a lot of time to find accurate values for the parameter initialization.

The next approch consisted in defining a class model by manual sampling. For each class, the user clics in the related part of the volume. The problem with this method is that you compute your mean value and variance using only a few samples. Your sample will never be bigger that one hundred points because it is not convenient. Then, your mean values and variances are not accurate. Moreover, results are not reproducible with this method. This the number of samples is reduced, means and variances can vary a lot with one more sample and you can never reproduce two times the same initialization.

Because of all these limitations, we proposed a new approach using a label map, to estimate each class model.

3.2.2 Method used

The idea is to create a label map. This map contains colors. There is one color for each class we want to segment. The relation color/class is stored in H (section), in the EM algorithm. This relation color/class is set up during the tree creation step DSFSDF.

The user creates a label map by coloring caracterisc regions for each tissue to segment, in the appropriate color. This gives a spatial information to the algorithm. It can now estimate

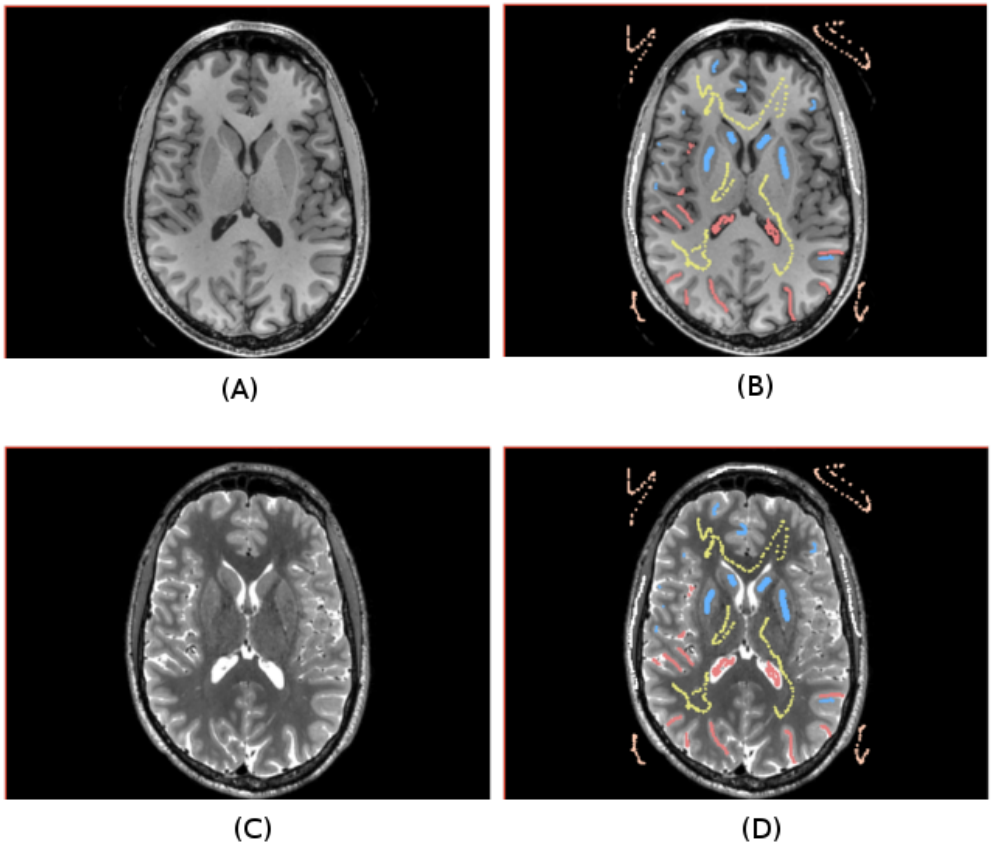

Figure 3.4: Axial view of the label map.

automaticly the mean value and covariance of each class, for each tissue, using this label map.

It is very convenient because, since the algorithm needs a good initialization, we can easily define a sample of hundred of points for each class. The results will representatives. Moreover, the results are now reproducible. Indeed, we can store then re-use the same label map. The results will remain the same.
NUMBERS TO PROVE DIFFERENCE!!!

3.3 Class Distribution visualization

An important contribution is a tool which allows to visualize the distribution of the classes to be segmented.

3.3.1 Interest

As discussed before, the algorithm is sensible to the initalization. It means that the initialization has to be good. Once the parameters are chosen, the user has no tools to know if his selection is accurate. Two classes to segment can't have too close means and variances. Even if the user sees the values he chooses, it is not easy to know if two classes to be segmented are too similar or not.

3.3.2 Our approach

The objective is to provide the user the most accurate and usefull vizualisation as possible.

We first assumed that each class has a normal distribution. We first decided to plot the gaussian in 3D, using the multivariate normal distribution. In the 2-dimensional nonsingular case,

the probability density function is

$$f(x,y) = \frac{1}{2\pi\sigma_y\sigma_y\sqrt{1-\rho^2}} \exp\Big(-\frac{1}{2(1-\rho^2)}\Big(\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}\Big)\Big)$$

(see [15]). x and y are the position of the pixel in the 2D space. $f(x,y)$ will return the value (heigth) of the (x,y) pixel. Each X and Y axe represent one volume. Let's first say that the range of the X and Y axes are the intensity range of the X and Y volumes. μ_x is the mean value of the class in the X volume. μ_y is the mean value of the class in the Y volume. σ_x and σ_y are the variance of the tissue in its respective volume. ρ is the correlation between X and Y. It indicates the strength and direction of a linear relationship between two random variables (see [16]).

We can easily deduce ρ from the covariance matrix Σ (see [17]). Indeed, in the 2D case, the covariance matrix can be expressed as

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}$$

The covariance matrix and the mean values for each class to segment for each image are computed during the labelmap sampling (section 3.2).

We said that the range of the X and Y axes are the intensity range of the X and Y volumes. The problem with this approach appears if the classes to segment are not spread over all the intensities. Indeed, the vizualisation is not good then: the gaussian is only localized in a small portion of the 2D plane. We want to "zoom" on the region of interest. We decided to change the range of the two axes. Then range will be re-defined for each image. Let's present it for a given image X. It is now the difference between Max, the maximum value extracted with the label map, between all the samples between all the classes for the image X, and Min, its opposite.

IMAGES

3.4 Intensity Normalization

Another very usefull contribution is a tool which helps the user to determine the good normalization value.

3.4.1 Interest

As discussed in section (2.5.1), at the step 4, an intensity normalization is done. Will already presented the utility of an intensity normalization in the same section. The problem is that the user has no tools to find the good values for the segmentation. He has to guess the mean intensity of the voxels in the MR image, background exluded. This is of course not doingable in practise.

3.4.2 Our approach

We implemented a simple tool, to allow the user to find easily and accurately this normalization value.

The first step of the work consited in creating the histogram associated to the image. The Y axe which presents the number of pixels for each intensity in the volume uses a log-scale because the range is huge. The log scale reduces considerably the range. We then added a cursor in this histogram. Using it, the user can choose the intensity which will be the "limit" between the background and part of interest. Finally, while the cursor is moving, our algorithm computes automatically the mean value of the voxels in the volume, from this intensity range to the highest intensity range. This is the normalization value.

We present in figure (3.5) the tool we developped. A T1 volume has been loaded. The user can move the cursor in the histogram. While moving, it returns in real-time the normalization value for the given position of the cursor in the lower frame.

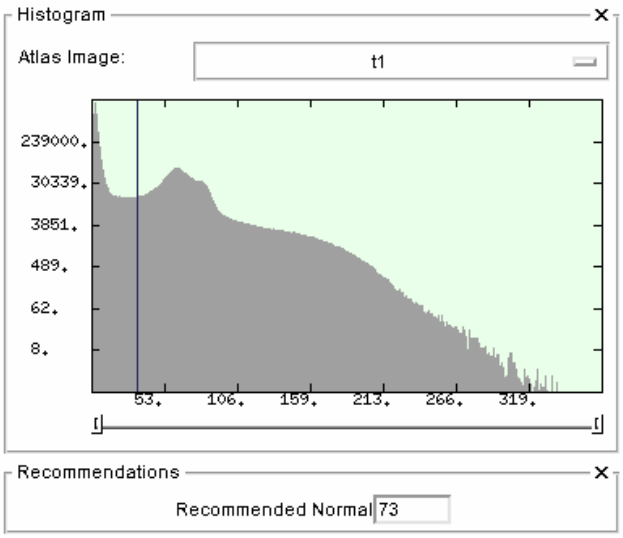

Figure 3.5: Tool developped for the intensity normalization parameter estimation

3.5 Global Prior Estimation

The last contribution to the EMS is a tool which provides the user an easy and fast way to estimate the global prior weights (GPW).(ref ch2 ...)

3.5.1 Presentation of the problem

This contribution is usefull in many different ways. When you run the segmentation process,at the 6th step of the process, you have to provide to the algorithm an estimation of the GPW for each node in the tree. First of all if there are a lot of structures to segment, they user can spend a lot of time during this step. Indeed, for each part of this tree strcuture, they have to define the GPW. Moreover, the user may not know at all which weights to choose. This new approach will provide the users a good estimations of the weights to use. We must also keep in mind that the end users are physicists. They might don't understand what the parameters meanings and providing them a visual feedback could help them a lot.

3.5.2 Our approach

We divided the problem in two parts. The first part will be about providing the user a real-time feedback regarding the GPW estimation. The second part will consist in developping an algorithm which fills automatically the tree.

Fast user feedback

We can divide the feedback part in 3 steps: the histogram computation and utilisation, the multicolumn list and the labelmap generated. The histogram allows the user to manual segment classes based on intensity. The multicolumn list allows the user to change the order of the classes in the histogram. The labelmap provides to the user a visual feedback, base on the segmentation realized in the histogram. Using these three complementary tools, the user, even if he is not initiated can estimate easily, accurately and rapidely the GWP.
schema

Global priors evaluation

The algorithm used to estimate the weight of each node is iterative. It starts from the root and goes to the leaves. It evaluates the weight of the childs of the active node at each iteration. Here is a description of the algorithms used to compute the GPW of each node.
DEscirption algo 1

Algorithm 1: TreeWeightEstimation(R, W)

---

define C = CHILD(R) ← set of childrens of root R
define LEAF(C) ← set of leaves of tree with roots C
define H ← set of structure-specific information defined by LEAF(C) for each leaf
update W in childrens of root R with the results of WeightEstimation(C,LEAF(C),H)
for each node R' in CHILD(R) that is not a leaf
▷TreeWeightEstimation(R', W)

Descripton algo2
The algorithm used estimates the global prior of the leaves of the current node, based on the number of pixel which belong to the child classes. This number of pixels is calculated from the segmentation computed in the histogram.(CF ..)

Algorithm 2: WeightEstimation(C,LEAF(C), W,H)

---

define T ← set of total weight of leaves in LEAF(C). Leaves weights are contained in H
define E ← set of weight for each node of C
for each node of C
▷E=E+H : Get the total weight of each node
W=E/T
return (W)

The new tool is presented in figure (A) (3.6). When the user moves a cursor, it changes the intensity range for each class. The figure (B) (3.6) is updated in consequence and provides a real time feedback to the user, about about what he is doing. Clicking on "update tree" (in (A)), the estimated size of a class is computed and is automatically fills the information into the tree structure (parameter H).

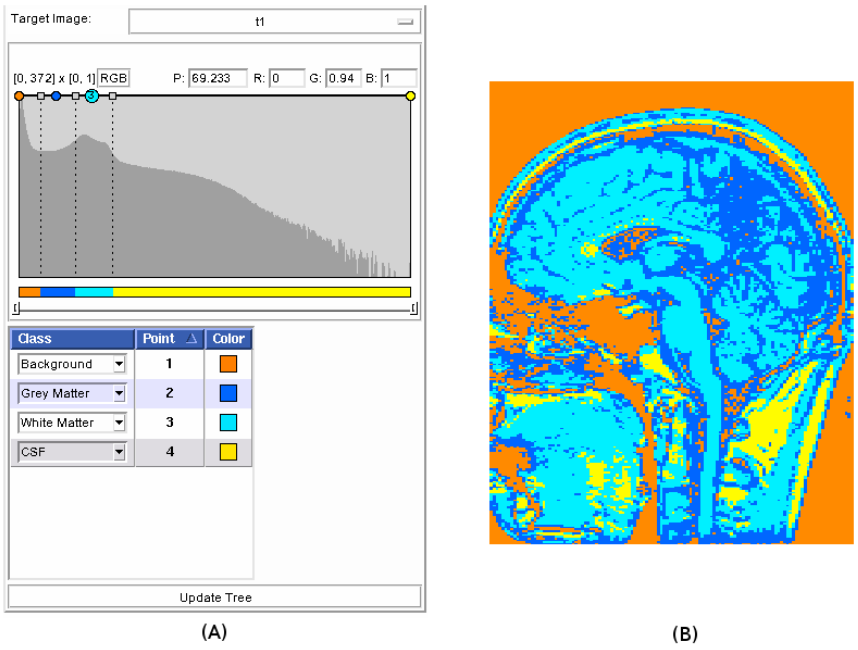

Figure 3.6: Tool developped for the global prior weights estimation

Chapter 4

Results and discussion

This chapter will present the importance of the contribution we did, regarding the final segmentation. We will not discuss about the enhanced usability of the module. It will allow you to see how the segmentation has changed. We will then discuss about the next contribution which could be brought to enhance the segmentation and the usability of a such module.

4.1 Results

Here we get going with a presentation of different results, using the different main contributions. We will conpare results obtained with the previous workflow and the new one. The results obtained will then be reviewed by a specialist to evaluate the segmentation. We always work with the same dataset. We chose intentionally a biased one.

4.1.1 Bias correction

Here we present the utility of the intensities inhomogeinities correction. It is usefull for the registration

Testing process

A tutorial for EM segmentation in Slicer 3 is available at SDFS. We followed it, one time with a biased image, one time with a correct image. All the parameters remains the same in the segmentation workflow.

Results

correction The results of the final segmentation are presented in figure (4.1). (A) presents the original biased image, (B) the corrected image, (C) the result of the segmentation with the biased image and (D) the result of the segmentation with the corrected image. The reults appears better but as long as we are not experts, an expert will estimate the result of the segmentation in the next section.

Nevertheless, we can explain the difference. It is due to a better registration and a better class distribution. QSDFQ More images with better resolution in annexes SFDSDF.

Experts's point of view

FFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFF

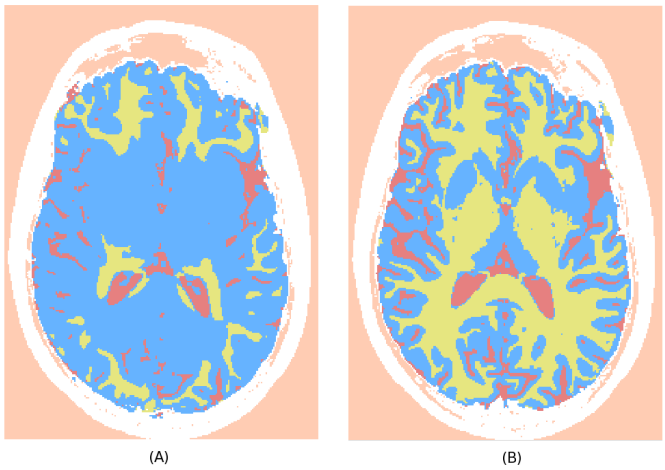

Figure 4.1: Results of the segmentation with bias correction

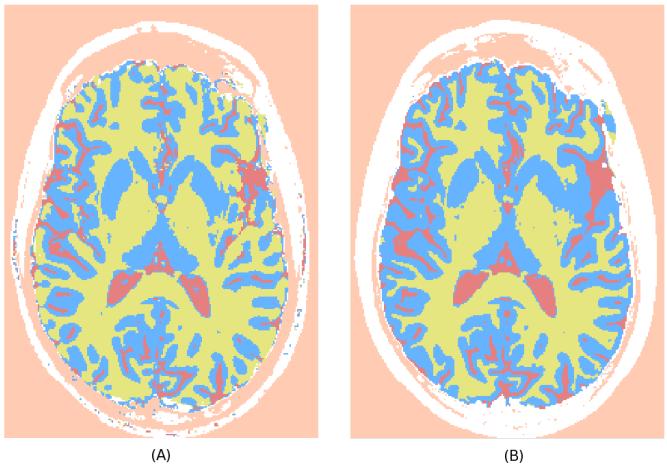

Figure 4.2: Results of the segmentation with label map

FFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

4.1.2 Class Selection

compare results regarding accuracy of distribution

Testing process

We followed the same tutorial as in the previous section. We used each time a corrected image

Results

underestimation overestimation holes filled

Expert's point of view

VVVVVVVVVVVVVVVVVVVVVVVVVVVVV

4.2 Future work

PRIOR A major issue of this approach is that when the image is biased and when some class have a close distribution. It is no longer effective because it is only based on the intensity of the voxels. Integrate the gaussian distribution in the estimation process.

Acknowledgements

Ron Kikinis who gave me the opportunity to carry out my intersnship in the SPL. Sylvain Jaume who supervises me during all my work. Steve helped a lot

Bibliography

[1] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the em algorithm", *Journal of the Royal Statistical Society: Series B*, vol. 39, pp. 1-38, November 1977.

[2] Y. Weiss, "Bayesian motion estimation and segmentation", *PhD thesis*, Massachussets Intitute of Technology, May 1998.

[3] R.C. Hardie, K.J. Barnard, and E.E. Armstrong, "Joint MAP registration and high-resolution image estimation using a sequence undersampled images", *IEEE Transaction on Image Processing*, vol. 6, pp. 1621-1633, December 1997.

[4] M. Murgasova, "Tutorial on Expectation-Maximization: Application to Segmentation of Brain MRI", May 2007.

[5] S. Borman, "The Expectation Maximization Algorithm",January 2009.

[6] Wikipedia, "Expectation-Maximization algorithm", *http://en.wiki.org/wiki/Expectation-maximization_algorithm*, June 2009.

[7] G. McLachlan, and T. Krishnan, "The EM Algorithm and Extensions", *John Wiley & Sons*, New York, 1996.

[8] K. V. Leemput, F. Maes, D. Vandermeulen et al., "Automated model-based tissue classification of MR images of the brain", *IEEE Transaction on Medical Imaging*, 18(10), pp. 857-908, 1999.

[9] K. V. Leemput, F. Maes, D. Vandermeulen et al., "Automated model-based bias field correction of MR images of the brain", *IEEE Transaction on Medical Imaging* 18(10), pp. 885-896, 1999.

[10] W. M. Wells III, W.E.L. Grimson, R. Kikinis et al., "Adaptative segmentation of MRI data", *IEEE Transaction on Medical Imaging* 15(4), pp. 429-442, 1996.

[11] K.M. Pohl et al., "A Hierarchical Algorithm for MR Brain Image Parcellation", *IEEE Transaction on Medical Imaging* 26(9), pp. 1201-1212, 2007.

[12] D. Hoa, and A. Micheau, "e-MRI, Magnetic Resonance Imaging physics and technique course on the web", *http://e-mri.org*, 2007.

[13] Kitware, "Insight Toolkit", *http://www.itk.org/*, 2009.

[14] N. Tustison, "Nick's N3 ITK Implementation For MRI Bias Field Correction", *http://www.insight-journal.org/browse/publication/640*, June 2009.

[15] Wikipedia, "Multivariate Normal Distribution", *http://en.wikipedia.org/wiki/Multivariate_normal_distri* June 2009.

[16] Wikipedia, "Correlation", *http://en.wikipedia.org/wiki/Correlation*, June 2009.

[17] Wikipedia, "Covariance matrix", *http://en.wikipedia.org/wiki/Covariance_matrix*, June 2009.

Appendix A

Statistics

Here we present the main formulas we used in this reports and some fundamentals about probabilities.

A.1 Fundamentals

$P(A)$ is the probability that A is realized.
$P(A|B)$ is the probability that A is realized, knowing B.
$P(A, B)$ is the probability that A and B are realized at the same time.
$P(A|B) = \frac{P(A,B)}{P(B)}$

A.2 Bayes' theorem

A.2.1 Theorem

Let S be a sample of space. If $A_1, A_2, ...A_n$ are mutually exclusive and exhaustive events such as $P(A_i) \neq 0$ for all i.Then for any event A which is a subset of $S = A1 \cup A_2 \cup ... \cup A_n$ and $P(A) > 0$ we have

$$P(A_i|A) = \frac{P(A_i)P(A|A_i)}{\sum_{j=1}^{n} P(A_j)P(A|A_j)}$$

A.2.2 Proof

We have $S = A1 \cup A_2 \cup ... \cup A_n$ and $A_i \cap A_j = \varnothing$ for $i \neq j$. Since $A \subseteq S$

$$\begin{aligned} \Rightarrow A &= A \cap S \\ &= A \cap (A1 \cup A_2 \cup ... \cup A_n) \\ &= (A \cap A_1) \cup (A \cap A_2) \cup ... \cup (A \cap A_n) \end{aligned}$$

Moreover

$$P(A \cap A_i) = P(A)P(A_i|A)$$

So

$$\begin{aligned} P(A) &= P(A \cap A_1) + P(A \cap A_2) + ... + P(A \cap A_n) \\ &= P(A)P(A_1|A) + P(A)P(A_2|A) + ... + P(A)P(A_n|A) \end{aligned}$$

And

$$P(A|A_i) = \frac{P(A \cap A_i}{P(A)}$$

Finally we obtain

$$P(A_i|A) = \frac{P(A_i)P(A|A_i)}{\sum_{j=1}^{n} P(A_j)P(A|A_j)}$$

A.3 Jensen's inequality

A.3.1 Inequality

Let f be a convex function defined on an interval I. If $x_1, x_2, ..., x_n \in I$ and $\lambda_1, \lambda_2, ..., \lambda_n \geq 0$ with $\sum_{i=1}^{n} \lambda_i = 1$,

$$f(\sum_{i=1}^{n} \lambda_i x_i) \leq \sum_{i=1}^{n} \lambda_i f(x_i)$$

A.3.2 Proof

To show that the theorem is true we proceed by induction.

- **Initialization**
  This is trivial for $n = 1$.

- **Hypothesis at rank** n

$$f(\sum_{i=1}^{n} \lambda_i x_i \leq \sum_{i=1}^{n} \lambda_i f(x_i))$$

- **Demonstration at rank** $n + 1$

$$\begin{aligned}
f(\sum_{i=1}^{n+1} \lambda_i x_i) &= f(\lambda_{n+1} x_{n+1} + \sum_{i=1}^{n} \lambda_i x_i) \\
&= f(\lambda_{n+1} x_{n+1} + (1 - \lambda_{n+1}) \frac{1}{1 - \lambda_{n+1}} \sum_{i=1}^{n} \lambda_i x_i) \\
&\leq \lambda_{n+1} f(x_{n+1}) + (1 - \lambda_{n+1}) f(\frac{1}{1 - \lambda_{n+1}} \sum_{i=1}^{n} \lambda_i x_i) \\
&= \lambda_{n+1} f(x_{n+1}) + (1 - \lambda_{n+1}) f(\sum_{i=1}^{n} \frac{\lambda_i}{1 - \lambda_{n+1}} x_i) \\
&\leq \lambda_{n+1} f(x_{n+1}) + (1 - \lambda_{n+1}) \sum_{i=1}^{n} \frac{\lambda_i}{1 - \lambda_{n+1}} f(x_i) \\
&= \lambda_{n+1} f(x_{n+1}) + \sum_{i=1}^{n} \lambda_i f(x_i) \\
&= \sum_{i=1}^{n+1} \lambda_i f(x_i)
\end{aligned}$$

With a concave function (in opposition to convex), the inequality becomes:

$$f(\sum_{i=1}^{n} \lambda_i x_i) \geq \sum_{i=1}^{n} \lambda_i f(x_i)$$

Appendix B

Results of the segmentation

Here we present the results of segmentations with the enhancement brought to the algorithm.

B.1 Label map sampling and biased target

The segmentation was processed without bias correction but using the label map for the definition of each class distribution.

B.1.1 Axial view

B.1.2 Coronal view

B.1.3 Sagittal view

B.2 Manual sampling and unbiased target

B.3 Label map sampling and unbiased target

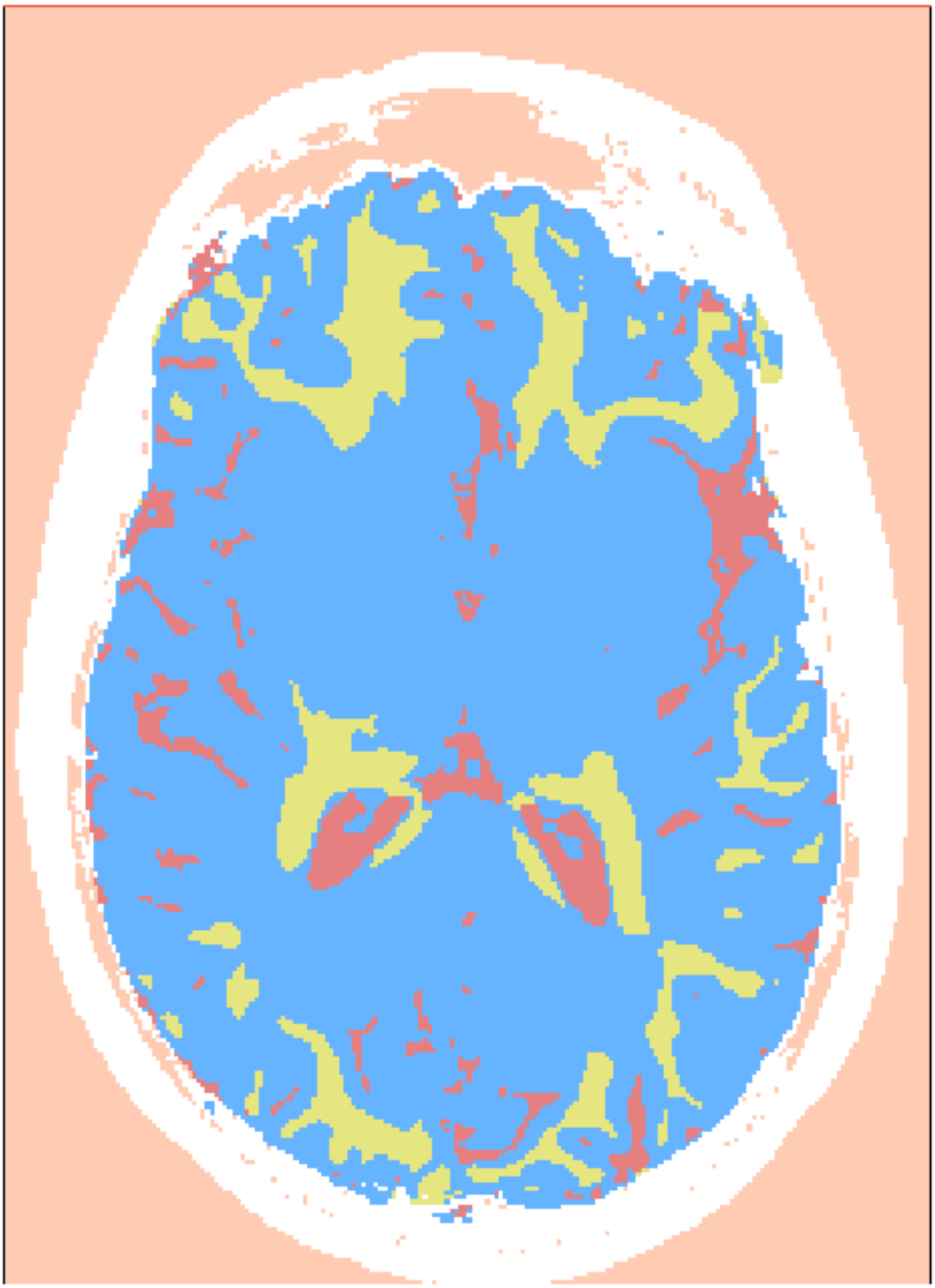

Figure B.1: Axial view of the segmentation's results with a biased target and a label map sampling.

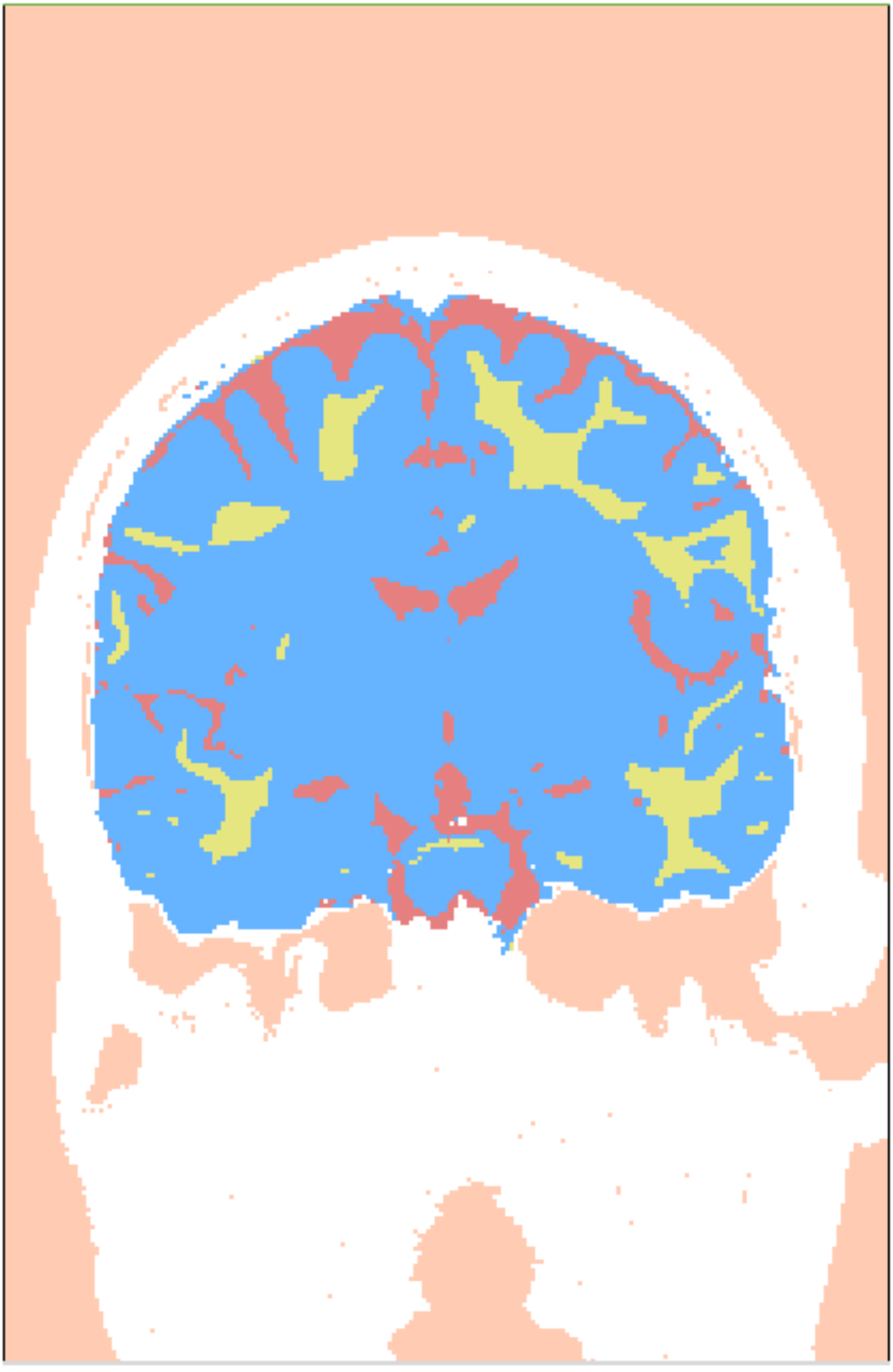

Figure B.2: Coronal view of the segmentation's results with a biased target and a label map sampling

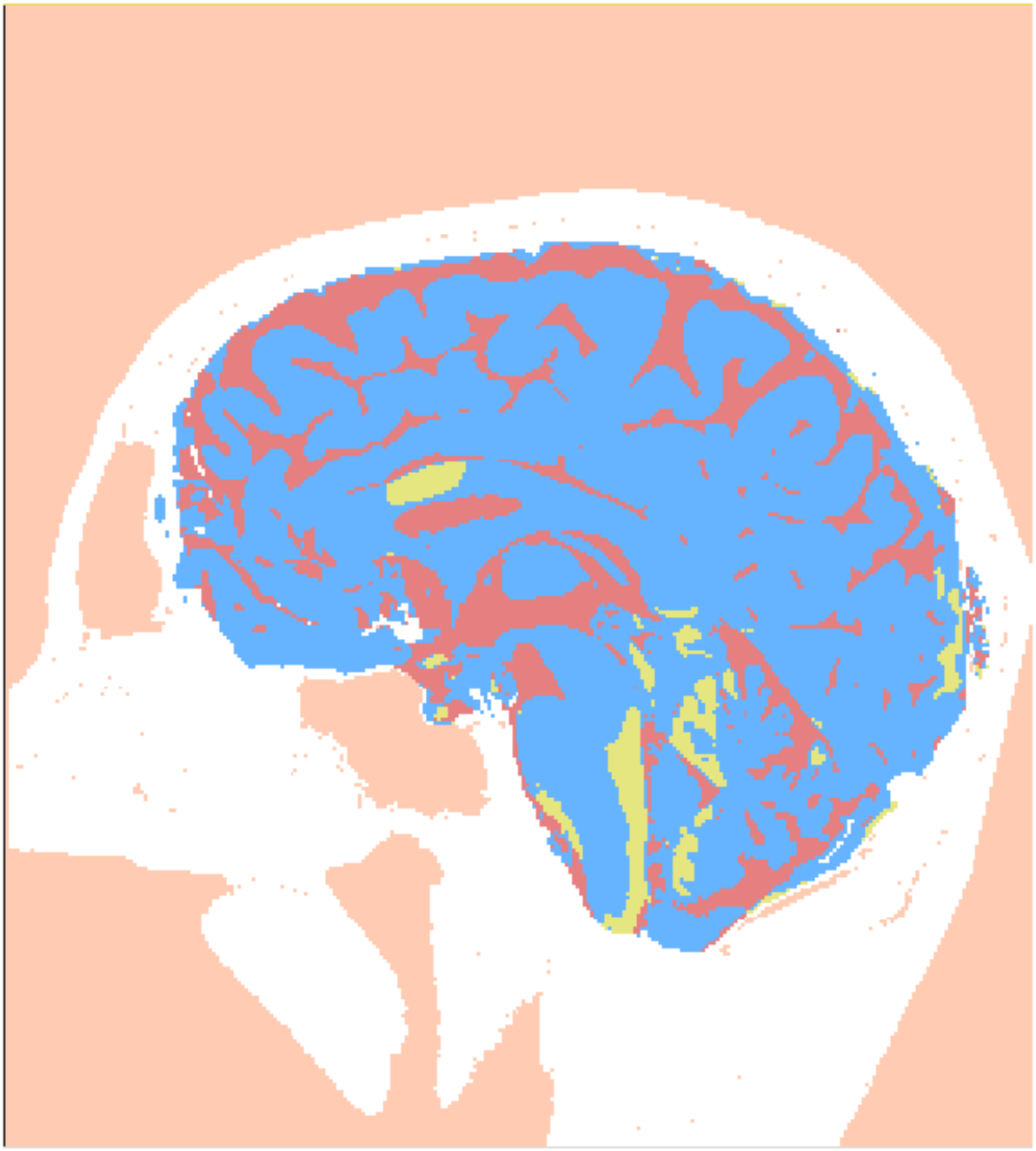

Figure B.3: Sagittal view of the segmentation's results with a biased target and a label map sampling.