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 Septemner 11th
At
Surgical Planning Laboratory (SPL)

SPL

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

Supervisors:

- Ron KIKINIS, Surgical Planning Laboratory, Harvard Medical School, Boston, MA - USA
- Sylvain JAUME, CSAIL, Massachusset's Insitut of Technologies, Boston, MA - USA

Referring teachers:

- Dominique MARATRAY, Institut Superieur de l'Electronique et du Numerique, Brest - FRANCE
- Christine CAVARO-MENARD, Universite de Angers, Angers - FRANCE

Student:

- Nicolas RANNOU, Institut Superieur de l'Electronique et du Numerique, Brest

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

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

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 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 $p(Y|\Phi)$.

As we said in SDFSDF, 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:

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

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

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

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

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

Let us denote

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

Equation (2.9) leads us to:

$$\mu_k=\frac{\sum_{i=1}^{n}y_ip_{ik}}{\sum_{i=1}^{n}p_{ik}} \qquad (2.11)$$

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:

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

$$c_k=\frac{1}{n}\sum_{i=1}^{n}p_{ik} \qquad (2.13)$$

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.

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.

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

  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. Most of the time, convergence is reached when the differnce between the estimation of the parameter Φ, at step m and step $m+1$ is smaller than ε:

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

We are no familiar with the theory behind the EM algorithm. The logic appears clearly and we can summarize the basic algorithm with the simple schema below:

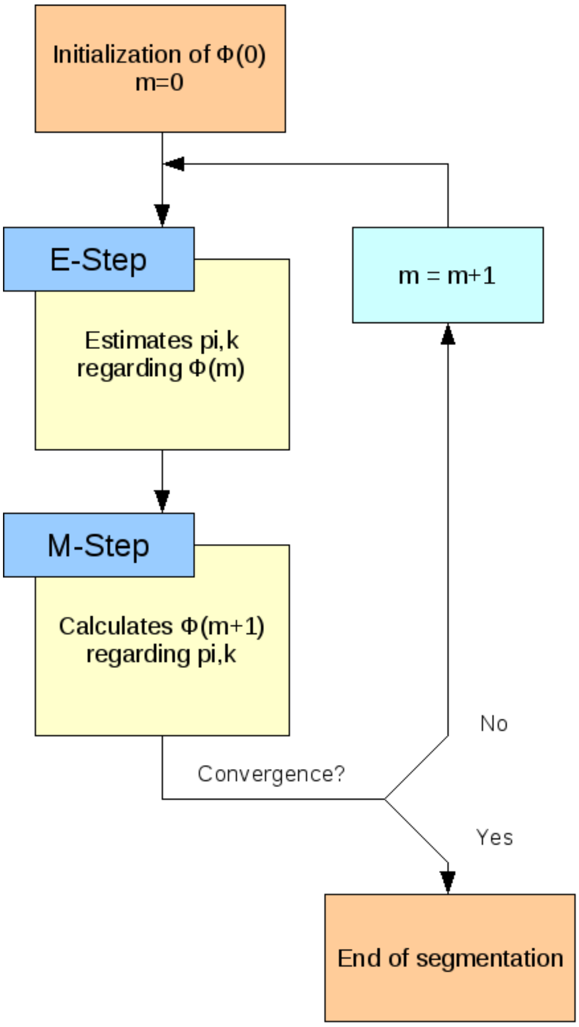


Figure 2.1: Basic EM algorithm

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 MR images are well suited to segment white matter (WM) but is really bad for cerebrospnial fluid (CSF). On the contrary, T2 MR images are well suited for CSF and not for WM. To formalize it, 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 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.

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 (ANNEXES) $\Psi_l(x)$. Let b_i be the realisation of the random variable B_i

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

$pos(i)$ 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 to estimate the probability that a voxel i belongs to class j. It will be computed during the M-Step, in parallel with the maximization of the parameters (mean and variance). 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 step can be interpreted as follows: the estimated soft segmentation and gaussian distribution parameters are used to reconstruct the image $\tilde{Y} = (\tilde{y_1}, ..., \tilde{y_n})$. The 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.

The approach used in Slicer 3 is based on the same principle but differs regarding the parameters which are known and estimated, in the EM algorithm.

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]) The EM is used to estimate the bias field using the *maximum a posteriori principle* (MAP) instead of the maximum likelihood principle (equation (2.6)):

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

Proceeding the same way 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.We also 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 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.

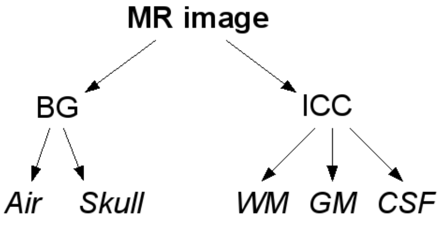


Figure 2.2: A simple tree structure of the brain

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). 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.

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

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 brain segmentation 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, given parameter $\Phi^{(m)}$.
- **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

This will be iterates until the whole tree has been processed as described in section 2.4.4.

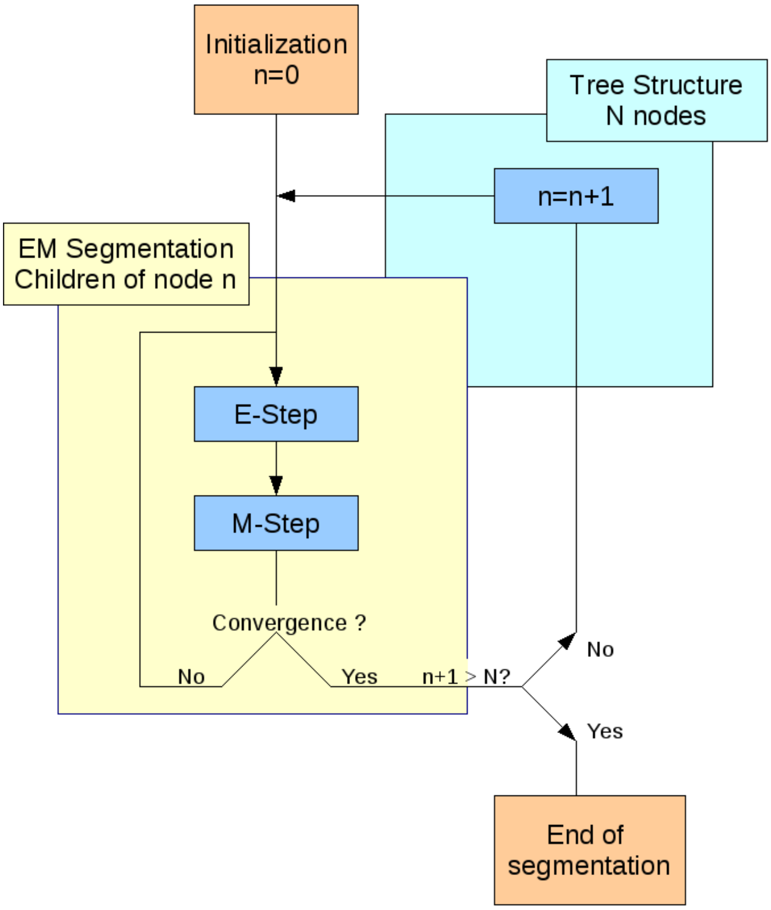


Figure 2.3: EM segment algorithm in Slicer

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 0, i.e., BG and ICC. Once the EM Segmentation has converged, BG and ICC are segmented and we go to the next node. Air and Skull will then be segmented. Following this process, all the structures of the tree will be segmented.

2.5 Workflow in Slicer 3

We will now present the whole segmentation 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 in the pre-processing treatments or in the EM algorithm.

2.5.1 User interface

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 for each node of the tree, i.e. each tissue to segment, the related atlas. It will be used for the spatial information (section 2.4.1).

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 because some tissues are not clearly visible in some modalities. It helps us to define Y ,the observed images. It will be used for the multichannel segmentation (section 2.4.2).

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 a more precise way to initialiase 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 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 as which modality is the more relevant for the class and the size that the class is supposed to be.

For example, for the CSF. If we give a weight of one (maximum) to the T2-target volume and zero 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.

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.

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 (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.

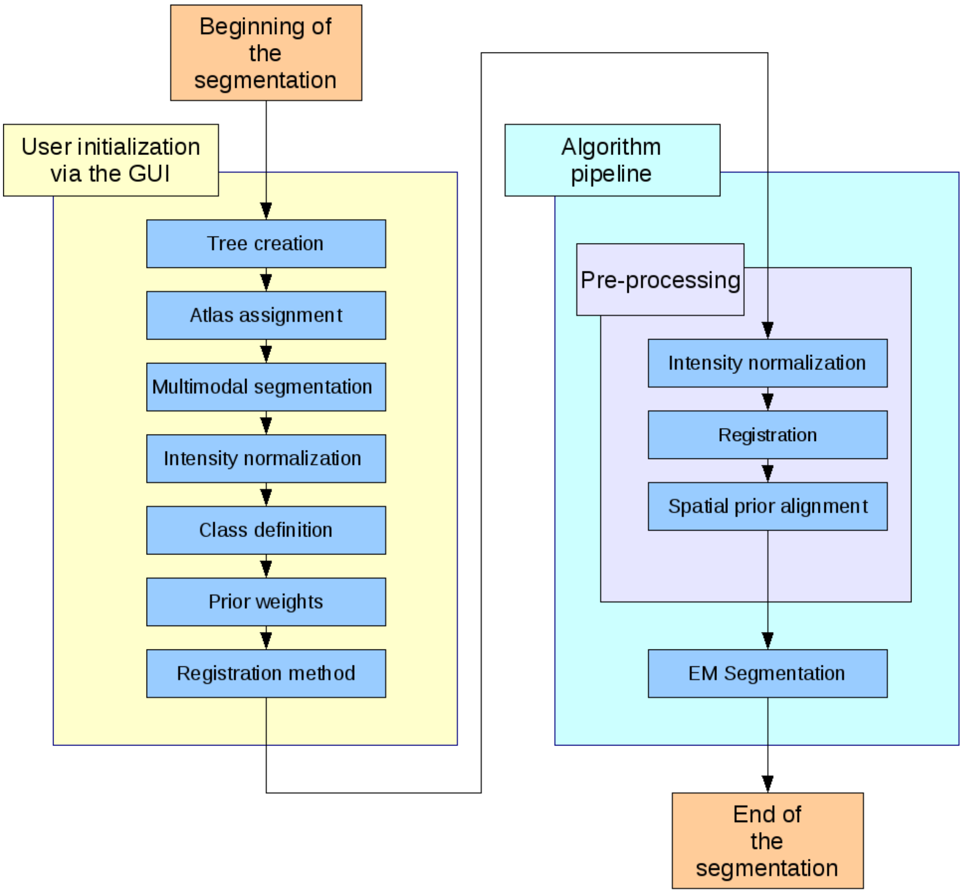


Figure 2.4: The whole segmentation pipeline in Slicer 3

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 in the intensity normalization step. The user has no tools to identify the good normalization values and 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 expect 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, to estimate the transformation. 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 try to solve the problems.

Chapter 3

The contributions

3.1 MRI Bias Field correction

Bias correction is not efficient so we propose an other method, as part of the segmentation process.
Rescaling

3.1.1 Interest

algorithm is only accurate for small region bias correction, because au the gaussian model he uses. Correction the bias in the whole volume first could enhance results.

3.1.2 Our approach

Gaussian distrib.... ITK filter Automatic

3.1.3 Results

corrected not corrected parameters explanation

3.2 Intensity Normalization

3.2.1 Interest

usefull if you need to do a lot of segmentations enhance the rescaling..?

3.2.2 Our approach

background/brain

3.2.3 Our implementation

show results... corrected image...

3.3 Class Distribution selection

During parameters initialization, the user has to define each class distribution. The current methods presents some troubles so we present here a new approach to estimate each class distribution.

3.3.1 Interest

So far, the user haf two choices to define each class distribution. He could enter 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 for the user to found the good mean value and variance for each class for each volume. Morever, each time he wants to process a new volume, he will have to redefine mean values and variances. This is not convenient and it can takes a ot of time. 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. With this methos, the problem is that you compute your mean value and variance using a few samples. Then, your mean values and variances are not accurate. That's why we proposed a new approach using a label map, to estimate each class model.

3.3.2 Method used

Create a label map. colors we choose a the same as the one we defined for each class. then run algorithm which will estimate the mean value and covariance values.

Algorithm..?

Explain each colors

3.4 Class Distribution visualization

An important contribution is a tool which allows to visualize the classes to be segmented distribution, through 2 volumes.

3.4.1 Interest

As we discussed in the EMS algorithm requires variance and mean values for each class to be segmented. So far, the user defines these parameters by selecting samples on the dataset. exampleSDFSF A main issue is that user

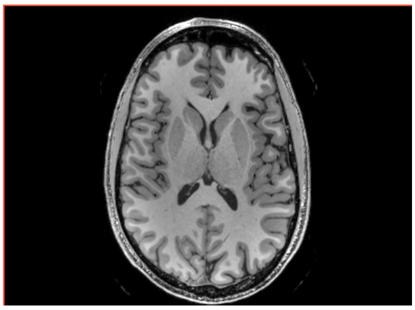


Figure 3.1: Axial view of a T1 volume without the label map.

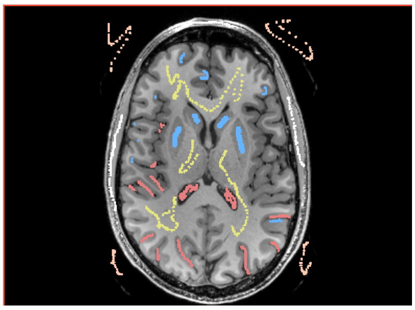


Figure 3.2: Axial view of a T1 volume with the label map.

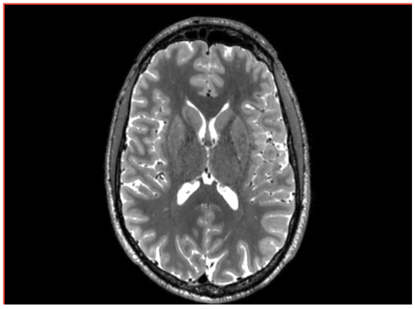


Figure 3.3: Axial view of a T2 volume without the label map.

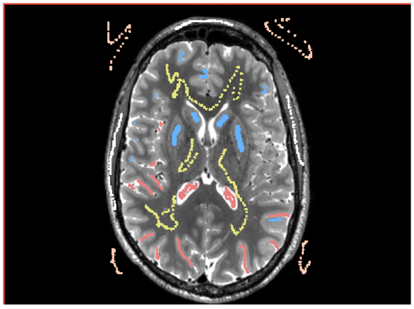


Figure 3.4: Axial view of a T2 volume with the label map.

user can know if the pixels he selected are really representative or not of the class he wants to select. Thus, our objective was to find a way to give the user if his samples are reprenstative of not of a tissue.

3.4.2 Method used

Problem's description

Here is a brief description of the situation to let you understanding the problem. As input we have N images, K classes to segment, and for each class a mean value μ, a variance σ and covariance ζ. The covariance is estimated for each couple of images. We will come back on it later. where Using these informations, we had to find a way to give the user usefull informations about the class distribution.

Choices

Our first idea was to display gaussian curves in 3D, for each class. The X and Y axes would be the intensity ranges for the to images we want to use. The problem with this representation is that it does not contain any information about the number o f pixels . 3d too sophisticated...? To solve this problem, we decided to divide the problem in two parts. The first part will consist in creating a "temperature map" as background. The second part will consists in displaying ellipses over this map. This will give the user a good information about how good his selection was.

Background creation

First of all, the background has to be created. This background is a "temperature map". We can present it as a two dimensions array. The height and the lenght of the array depend of the intensity range of the 2 volumes we want to process. Then, we fill the array in an easy but useful way. Each case of the array contains the number of occurence of an intensity pair at the same position, through the 2 volumes.

Algorithm 1: BACKGROUNDCREATION(V1,V2)

```
define RangeI1 ← intensity range in V1
define RangeI2 ← intensity range in V2
define A(RangeI1,RangeI2) ← background
for each pixel i in V1
for each pixel j in V2
▷A(i,j)=A(i,j)+1
return A
```

Ellipse representation

We will first start we a brief rappel.. The covariance matrix is
Representing ellipse associated to this covariance matrix.
Maximum variance axe.Eigen values
Algorithm used. Big and small axes of the ellipse

3.4.3 Results

show result

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)
```

3.5.3 Results and discussion

As we can see, the results obtained are good and accurate.The time of processing is fast and most of the people are now able to understand these parameters, even they are not familiar with EMS. This is an intuitive way

to get an estimation of the GPW parameters.
Of course some sfsdfs must be done. If there is a strong bias in the images we process, the pixels of a same class will have different values. Then, a segmentation based on intensity will provide bad results. Moreover, if the user wants to segment 2 classes which have the same color, just using the spatial information for example, they won't be able to use it properly.Thus, the user must always keep in mind that it is just an estimation and that he has to check if the values are accurate.

Chapter 4

Results and discussion

This chapter will review what has been done and mentions the main open questions.

Results using different tools

limitations

improvement (in relation to limitations)

4.1 Results

Tests has been performed to show the utility of the work done. The results are presented below. The testings are comparaisons between the results obtained with the previous workflow and the new one. The results obtained will then be reviewed by a specialist to evaluate the previous and the new segmented datasets.

4.1.1 Bias correction

Here we get going with the intensities inhomogeinities correction.

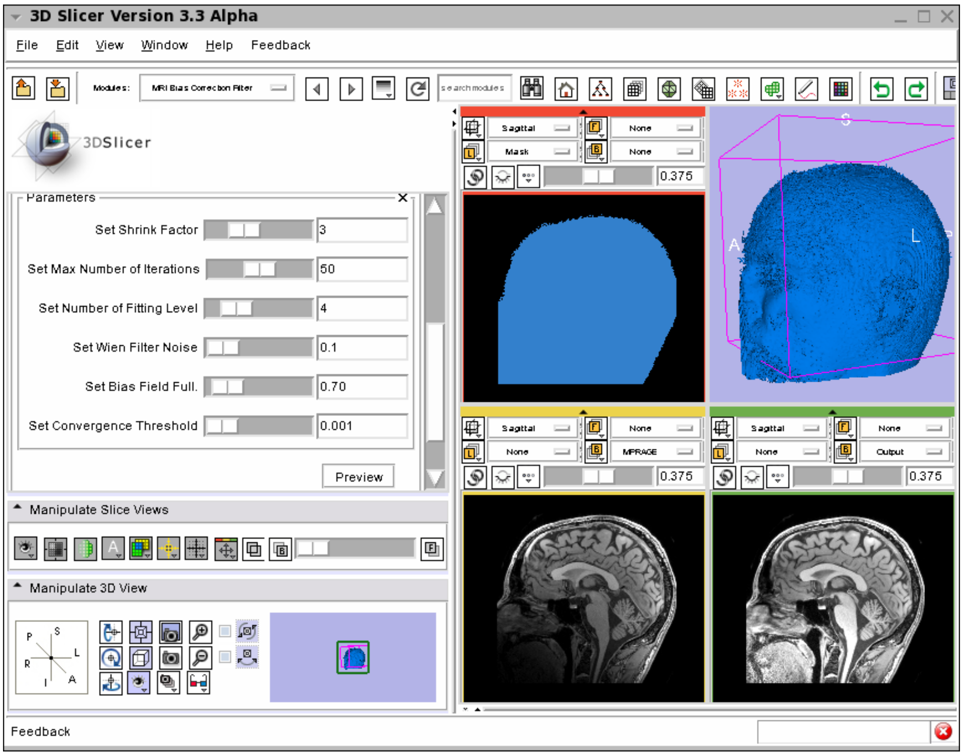
Testing process

we create a mask.

Results

Here are the resultsfdd dddddddddd ddddddddddd dddddddddd dddddddd ddddddddd dddddddddd ddddddddd ddddddddd dddddddd ddddddddd dddddddd ddddddd ddddddd dddddd dddd dddddddd..

more comments tttttttttttttttttttt tttttttttttt tttttttttt ttttttttttt tttttttttt tttttttttt ttttttttttttt tttttttttttt ttttttttttttt ttttttttttt tttttttttttttt



3D Slicer Version 3.3 Alpha
File Edit View Window Help Feedback
Modules: MRI Bias Correction Filter
search modules
3DSlicer
Parameters

Set Shrink Factor 3

Set Max Number of Iterations 50

Set Number of Fitting Level 4

Set Wien Filter Noise 0.1

Set Bias Field Full. 0.70

Set Convergence Threshold 0.001

Figure 4.1: Module created for the bias correction.

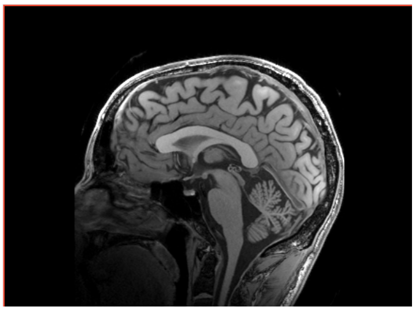


Figure 4.2: Sagittal view of a biased T1 volume.

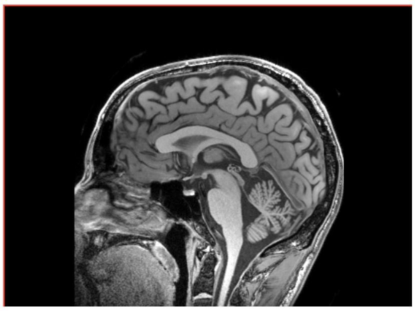


Figure 4.3: Sagittal view of the T1 volume after bias correction.

Specialist's point of view

4.1.2 Global Prior estimation

Testing process

Results

Specialist's point of view

4.1.3 Class Selection

Testing process

Results

Specialist's point of view

4.2 Limitations

4.3 Future work

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. Andryi, Daniel, Steve?

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.

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:

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