

PARAPET

Specification of Bayesian Reconstruction Building Blocks

Document: D/5.4/GENEVA/LABBE-SAUGE-MOREL/May 19, 2010/1/0.1

/TECHNION/JACOBSON/

/MILAN/PAGANI-VALENTINO-GILARDI/

DRAFT

Approved by:

C. Labbé V. Friedrich

(Author) (Project manager)

I. INTRODUCTION	1
II. MAXIMUM A POSTERIORI (MAP) ALGORITHMS	1
II.1. A brief Note on Maximum Likelihood - Expectation Maximization Algorithm (ML-EM)	1
II.2. Basic Principle of MAP Reconstruction Algorithm	2
II.3. One-Step-Late Algorithm (OSL)	2
II.4. Markov Random Field Algorithm (MRF)	4
II.5. Gibbs Random Fields Algorithm (GRF)	5
II.5.1. Introduction	5
II.5.2. Algorithm	5
II.5.3. Potential functions	7
II.6. Median Root Prior Algorithm (MRP)	9
II.7. A Modified OSEM Reconstruction Method based on Partial Volume Effect Correction Techniques (IPVE)	9
II.7.1. Partial Volume Effect	9
II.7.2. Algorithm	10
II.7.3. Conditions	13
III. COMPARISON OF MAP RECONSTRUCTION WITH OSEM AND ANALYTIC RECONSTRUCTION ALGORITHMS	13
III.1. Results On Hoffman Brain Phantom Through Reconstructed Images	13
III.2. Results On Utah Phantom Trhough Mediane Profiles	14
IV. MAP BUILDING BLOCKS	17
IV.1. MAP Method Class	17
IV.2. MAP Reconstruction Parameters Class	18
IV.3. MAP Reconstruction Class	19
V. SEGMENTATION	24
V.1. Segmentation by User's Defined Thresholds	24
V.2. Automated Segmentation by Fuzzy C-Means Method	24
V.2.2. Method	24
V.2.3. FCM algorithm	25
V.3. Processing of The Image Before Segmentation	26
V.4. Class of Segmentation Process	27
VI. REFERENCES	28

I. INTRODUCTION

Recently, there has been considerable interest in developing statistical computationally efficient tomographic reconstruction algorithms which produce the best possible reconstruction quality and accuracy. However, an undesirable property of the maximum likelihood reconstruction-expectation maximisation (ML-EM) (see D5.2 for more details) algorithm lies in the fact that after a certain number of iterations, the reconstructed images become increasingly noisy.

Hence, the development of scanners combining both anatomical and functional information is motivating the development of Bayesian maximum *a posteriori* (MAP) reconstruction techniques. The morphological information extracted from an X-ray computed tomography (CT) or magnetic resonance (MR) imaging may be used as a prior information in the reconstruction of the PET images [Ardekani et al., 1996].

Bayes theorem allows to introduce in the reconstruction process a prior distribution that describes properties of the unknown image. Maximisation of this *a posteriori* probability over the set of possible images results in a MAP estimate. Priors may be added one by one into the estimation process, assessed individually, and used the guarantee a fast working implementation of preliminary versions of the algorithms [Lalush et al., 1995; Lange et al., 1990]. The objective of this task is to detail the principles of different MAP reconstruction methods and to describe their specific building blocks which extend the collection of ML-EM building blocks developed in D5.2.

II.1. A brief Note on Maximum Likelihood - Expectation Maximization Algorithm (ML-EM)

The Maximum Likelihood - Expectation Maximization (ML-EM) has been described in details in D5.2. Here's briefly the MLEM formula which is given by:

$$\lambda_{i}^{MLEM(k+1)} = \lambda_{i}^{k} \sum_{i=1,..,N} \frac{n_{d} p_{id}}{\sum_{i'} \lambda_{i'}^{k} p_{i'd}} = \lambda_{i}^{k} c_{i}^{L(k)}$$
 (Eq. 1)

where n is the measured data, d the detector index, i the pixel index, p(i,d) the projection weighting factors, and λ^k the image of the kth iteration.

The ML-EM method updates the kth image lambda (λ^k) by a coefficient $c_i^{L(k)}$, which is calculated by comparing the measured data n and the reprojection of the current image. However, it is generally more difficult to find the optimum reconstruction as compared to the ML-EM algorithm.

II.2. Basic Principle of MAP Reconstruction Algorithm

ML-EM reconstruction of PET images is generally unstable (see D5.2), and the resulting images usually exhibit large variances. Maximizing over the posterior probability rather than over the likelihood function has a regularizing effect on this instability.

In the derivation of the EM algorithm, one tries to maximize the probability $p(y|\lambda)$, or $P(\text{coincidence data} \mid \text{image})$, of observing the measured data, given a current estimate of the activity distribution in the source. In terms of the unobservable latent data n and the observed counts y, the posterior probability $p(\lambda|y)$, or $P(\text{image} \mid \text{data})$, is given by the Bayes theorem as follows:

$$p(\lambda \mid y) = \int p(\lambda \mid n, y) p(n \mid y) \partial n$$
 (Eq. 2)

From a Bayesian point of view, the conditional posterior density $p(n \mid \lambda, y)$ can be expressed as the product of the likelihood function by a prior density function.

where p(y) is a normalisation constant and $p(\lambda)$ the prior probability. Calculation of an estimate of λ using this formula requires knowledge of both the predictive density p(n|y) and the conditional posterior density $p(\lambda|n,y)$.

The likelihood function based on n can be defined as the product of independent Poisson distributions. On the other hand, the prior density function has to be specified by the a priori information that one intends to incorporate into the reconstruction process. The prior reflects our expectations concerning the relative probabilities of different images. Now, the most probable image, given the set of measured data, can be obtained by maximising the right side of Eq. 3, i.e. the log of the posterior probability, as follows:

$$\lambda_{MAP} = \arg\max \left\{ \ln[p(\lambda \mid y)] = \ln[p(y \mid \lambda)] - \ln[p(\lambda)] \right\}$$
 (Eq. 4)

Note that all the following MAP algorithms are in fact a modified ML-EM algorithm in which a prior model has been incorporated in order to yield better resolution, recovery contrast, uniformity, etc.

II.3. One-Step-Late Algorithm (OSL)

As an example to overcome the problem of finding the optimum reconstruction in ML-EM, Green et al. (1990) proposed the One-Step-Late (OSL) algorithm where near optimum solutions of the Bayesian formulation of the problem can be obtained by modifying the general form of ML-EM (Eq. 1) by a multiplicative factor using the derivative of an *energy function*. The OSL-type Bayes algorithm modifies ML-EM by incorporating prior information as follows:

$$\lambda_i^{OSL(k+1)} = \frac{\lambda_i^{MLEM(k+1)}}{1 + \beta \frac{\partial}{\partial \lambda_i} U(\lambda, i) \big|_{\lambda = \lambda^k}} = \lambda_i^k c_i^{L(k)} c_i^{P(k)}$$
(Eq. 5)

where $U(\lambda)$ is defined as the energy function and β is an hyper parameter that influences the degree of smoothness of the estimated images. In this OSL formula, the second coefficient $c_i^{P(k)}$ should penalised only unwanted features of the image. Indeed, if the weight β of the prior is big, the resulting image follows closely the prior assumptions.

$$\frac{\partial U(\lambda)}{\partial \lambda_i}\bigg|_{\lambda=\lambda^{old}} = \sum_{k \in N} w_{ij} \frac{\partial V(\lambda_i - \lambda_k)}{\partial \lambda_i}\bigg|_{\lambda=\lambda^{old}}$$
(Eq. 6)

Actually, if $U(\lambda)$ is a roughness term, the image will be very smooth and will correspond to a low-pass version of the true image. If $U(\lambda)$ controls the edge sharpness based on an anatomical image, the final image will rather look like the anatomical image.

The energy function depends on the prior distribution assumed for the image and results in smoothing the image by scaling λ_i 's down when their values are higher than those of their neighbours and by scaling it up otherwise. The smoothing is the same throughout the image regardless of the position of λ_i . While this method generally yields better noise properties than ML-EM, the reconstructed images have lower resolutions and increased partial volume effects. To overcome this problem, some other penalty functions can be used which allow adjustment of the smoothing strength depending upon prior anatomical information on the image, or informations derived during the iterations.

The OSL algorithm shown in Eq. 5 will increase the value of the log-posterior probability at each iteration (Lange 1990). At least, it will therefore approach eventually at least a local maximum of the log-posterior function. The shape of the potential function $V(\lambda)$ will determine whether the log-posterior functions succeeds in mapping local maxima within the considered neighbourhood.

In the OSL equation, the neighbouring pixel value λ_k^{old} does not change with λ_i and can be considered as a constant regarding the partial derivative when computing the energy derivative term. We will then substitute $\lambda_i - \lambda_k$ by x so that x varies with λ_i . Since $\partial x/\partial \lambda_i = 1$, we can write the partial derivative in equation 6, in terms of x as follows:

$$\frac{\partial V(\lambda_i - \lambda_k^{old})}{\partial \lambda_i} = \frac{\partial V(x)}{\partial x}$$
 (Eq. 7)

Equation 7 applied to Eq. 6 shows that the derivative of the energy function is nothing but the sum of the N computations of the derivative of the potential function V that has to be defined for each pixel. In order to perform a reconstruction using the MAP-OSL algorithm, it is therefore necessary to determine the parameter β and the derivative potential function (DPF) $\partial V(x)/\partial x$. In section II.5, we will illustrate how the DPF determines how smoothing is distributed among intensity differences in the image.

II.4. Markov Random Field Algorithm (MRF)

Markov Random Fields (MRF) are used to describe the relationship between adjacent pixels. First of all, let define a neighbourhood system. The first order neighbours of a given pixel comprises all the closest pixels to the given pixel. They are all 1 pixel unit away from the given pixel. The second order neighbours of a given pixel add to the first order neighbours all the diagonal pixels that are at most 1/Ã2 pixel units away from the given pixel (Figure 1).

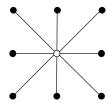


Figure 1: 2D neighbourhood system

Let N_k define a neighbourhood for each pixel $k \in W$. The set $N_W = \{N_k : k \in W\}$ defines the neighbourhood system.

Consider a random field defined on the lattice of pixel centres, with value at pixel k denoted X_k . Let X be the vector of pixel values $(X_1,...,X_k)$. The random field X is a MRF with respect to a neighbourhood system N_W , if and only if:

- (a) $p(X_k = x_k \mid X_i = x_i, i \in W \setminus \{k\}) = p(X_k = x_k \mid X_i = x_i, i \in N_k)$
- (b) $p(X = x_k) > 0$ for all possible configurations $x = (x_1,...,x_k)$.

Here \ denotes set exclusion, so $A \setminus B$ is the set of elements in A but not in B.

Condition (a) is known as the Markov property. It states that the probability of X having a certain value at a particular pixel, given the values of X elsewhere, is dependant only on the value of X in the neighbourhood of that pixel.

The positivity condition (b) states that all combinations of pixel values are possible. If in addition, the probability of the field taking any value at a given pixel, given the values at the neighbouring pixels, is independent of that particular pixel under the condition (c) expressed below $\forall k \in W, \ \forall i+j \in W, \ \text{and} \ \forall x_k$, then the MRF is homogeneous.

(c)
$$p(X_k = x_k \mid X_i = x_i, i \in N_k) = p(X_{k+1} = x_k \mid X_{i+1} = x_i, i \in N_k)$$

Early work with MRFs was hampered because it was not known how to evaluate the joint probability distribution p(X = x), or even how to define the local conditional probabilities such that the joint probability distribution was valid. These problems were solved by identifying MRFs with Gibbs random fields.

II.5. Gibbs Random Fields Algorithm (GRF)

II.5.1. Introduction

Gibbs Random Fields (GRF) originate from statistical physics, where the problem is to estimate large scale properties of a lattice system from its local properties. Ising (1952) pioneered the approach for modelling the behaviour of ferromagnetic material by considering only the spin interactions of neighbouring atoms. Two neighbouring atoms of opposite spins were considered to have a positive potential. The state of the lattice was characterised by its energy, computed as the sum of the potentials. Configurations of low energy were therefore more stables than those of high energy.

Application of this concept to image reconstruction is somewhat easier to understand if the ferromagnetic case is kept in mind. Thus, the Bayesian model can utilise a Gibbs prior to describe the spatial correlation of neighbouring regions like it was first suggested by Geman et al. in 1984, or it can also take into account the effect of limited spatial resolution.

II.5.2. Algorithm

A random field X is a Gibbs random field if and only if the probability (density) function has the following form:

$$p(X = x) = \frac{e^{-U(x)}}{Z}$$
 (Eq. 8)

Here again, $U(\lambda)$ is called the *energy function*. The higher is the energy of the configuration, the lower will be the probability. The denominator Z is the partition function, a normalisation constant obtained by summing the numerator over all possible configurations x taken by X. The partition function is usually not computable. For example, a small field of $K = 64 \times 64$ pixels, each of which taking only two values, has 2^{4096} possible configurations. The energy function can be specified in terms of potentials for a particular neighbourhood system N_k by:

$$U(k) = \sum_{k \in N_k} V(k)$$
 (Eq. 9)

A wide range of MRF priors using Gibbs distribution have been studied for emission and transmission. This includes intensity processing models (Geman and McClure 1987, Bouman and Sauzer 1993, Green 1990, Hebert and Leahy 1989), and compound models utilising line processing (Johnson et al. 1991). The contribution of these models to

image reconstruction can be summarised as the introduction of a coupling term generally modelled by a Gibbs prior distribution that will influence the creation of edges in the vicinity of significant anatomical edges.

Hence, a MRF model for a class of images can be fulfilled by a Gibbs distribution like the a priori probability of the activity concentration λ described by:

$$p(\lambda) = \frac{1}{Z} \exp[-\beta U(\lambda)]$$
 (Eq. 10)

where $U(\lambda)$ is the energy function, Z the partition function, and β an hyper parameter that influences the degree of smoothness of the estimated images. This latter parameter represents somehow a measure of the strength of the relationship between the pixels. The energy function can be written as:

$$U(\lambda) = \sum_{i} \sum_{j \in N_i} k_{ij} V_{ij} (\lambda_i - \lambda_j)$$
(Eq. 11)

where V_{ij} are pairwise potential functions defined on a 8-nearest-neighbourhood system for 2D images, and a 26-nearest-neighbourhood system for 3D images. k_{ij} are constants which represent the inverse of the Euclidean distance between the centres of each pixels, and λ_i and λ_j are intensity values of neighbouring pixels which belong to the same anatomical region. For neighbouring pixels belonging to different anatomical regions, $V(\lambda)$ is set to zero.

The energy function of the Gibbs prior is thus determined by a set of parameters with the intent to smooth noise and keep edge informations. The Gibbs prior is controlled by three parameters: one is determining the overall weight placed on the prior in the reconstruction process, and the two others are affecting the relative smoothing of noise versus edges in the image estimates. Usually, these potentials are symmetric increasing functions of the intensity difference between neighbouring pixels.

In image reconstruction, prior information is particularly useful. A major advantage of these probabilistic models is that global properties of the image can be modelled through its local properties. This means in practice that the potential function is defined only on the local neighbourhood of each pixel in the image lattice. Although there is some freedom in the choice of the size of the neighbourhood, many of the previously proposed Gibbs priors typically exploit only pairwise interactions of nearest neighbours and the use of higher order interactions has not been much investigated yet. It has been observed however that most of pairwise-interacting priors are often inadequate.

In this task, different Gibbs prior potentials are studied for a 3D local neighbourhood in which 3×3 pixel cliques, and pair cliques, are used to model the continuity of borders around image regions, and the smoothness within individual regions, respectively. However, the reconstruction problem expressed in a Bayesian framework is difficult to solve because of the non-convexity and non-linearity introduced by the prior itself. Therefore, stochastic sampling techniques are employed to solve this problem. The mechanism involved is essentially a random walk in the discrete configuration space where pixels are iteratively updated in a scanline order.

II.5.3. Potential functions

The derivative potential function (DPF) will act as a penalty term that enforces conditions required by the prior. The faster the potential V(x) increases with x (i.e. with the pixel intensity difference between λ_i and λ_k), the bigger will be $\partial V(x)/\partial x$ and hence the larger will be the penalty applied to λ_i . This results in minimising the difference between neighbouring pixels bounded by a sharp raising potential function (Figure 2).



Figure 2: Example of the application of a Green prior.

To illustrate the above property, we will use four priors previously described in the literature (Rapha'l Thierry, PhD Thesis, Grenoble, 1999). These are the priors of Geman and McCLure (1985) (GM), Green (1990) (GR), Hebert and Leahy (1989) (HL), and hypersurface (HS). These potentials are defined with respect to the variable $x = \lambda_i - \lambda_k$ (Table 1).

The energy functions and their derivatives are plotted in Figure 3. The different behaviours of these potentials is quite apparent in figure 3 b: the y-axis represents the strength with which the algorithm will try to suppress pixel intensity differences expressed along the x-axis.

Potentials	V(x)	Convex	V'(x)/2x	Trend
GM: Geman and McClure	$V_{GM}(x) = \frac{x^2}{1 + x^2}$	no	$\frac{1}{(1+x^2)^2}$	$\frac{1}{x^4}$
GR: Green	$V_{GR}(x) = 2\log[\cosh(x)]$	yes	$\begin{cases} 1, x = 0 \\ \frac{\tanh(x)}{x}, x \neq 0 \end{cases}$	$\frac{1}{x}$
HL: Hebert and Leahy	$V_{HL}(x) = \ln(1+x^2)$	no	$\frac{1}{(1+x^2)}$	$\frac{1}{x^2}$
HS: Hypersurface	$V_{HS}(x) = 2\sqrt{1 + x^2} - 2$	yes	$\frac{1}{\sqrt{1+x^2}}$	$\frac{1}{x}$

Table 1: Potential functions are given together with some of theirs properties

The advantage of using priors with peaked DPFs, such as GM and HL, is given by the fact that, if the bulk of pixel intensity differences due too noise fall under the peak, and larger edge differences fall on the right of the peak, noise can be smoothed without degrading edge information of the image. This intensity-selective smoothing operates

directly in the spatial domain and offers a significant advantage over linear filtering, which smoothes both noise and edges.

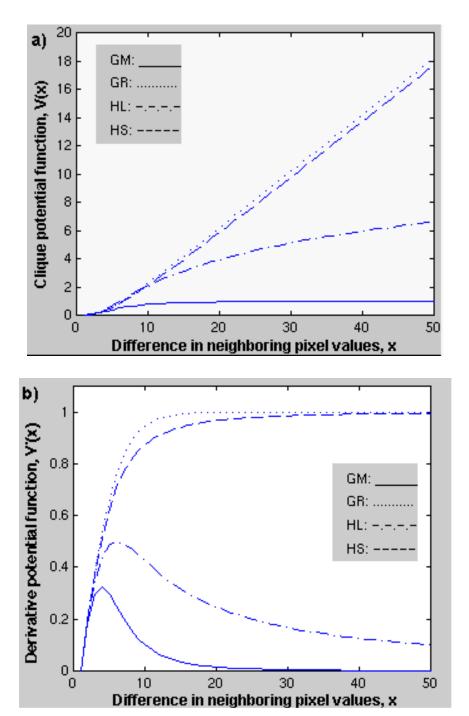


Figure 3: *Plots of four potentials (a) and their derivatives (b).*

II.6. Median Root Prior Algorithm (MRP)

A recently MAP method (Alenius, 1998) based on OSL-type algorithms, makes use of a median root prior (MRP) for $p(\lambda)$. The MRP assumes that the most probable value of the pixel is close to the local median. For this, the ML-EM update $\lambda_i^{MRP(k+1)} = \lambda_i^k c_i^{L(k)}$ is multiplied by a weight as follows:

$$\lambda_i^{MRP(k+1)} = \frac{\lambda_i^{MLEM(k+1)}}{1 + \beta \frac{\lambda_i^k - med(\lambda, i)}{med(\lambda, i)}} = \lambda_i^k c_i^{L(k)} c_i^{P(k)}$$
(Eq. 12)

where $med(\lambda, i)$ is the median of image pixels over a neighbourhood around pixel i. In this case, the penalty is set with respect to how much the pixel deviates from the local median $med(\lambda, i)$. Hence, no penalty at all is applied at edges. If β is large, sharp variations of the image estimate like isolated spikes or single pixel wide lines will be discarded and the image estimate will look like locally very monotonous. Given the spatial size of the median mask, the size of the smallest spatial detail preserved by the prior is then known.

With smoothness priors, the problem is that image and noise are overlapping in the frequency domain. This is the reason why we cannot distinguish the image from the noise by frequency selective low-pass filters. With MRP, the image and the noise can be distinguished on the basis of monotony. In a Bayesian sense, the prior assumption says that the true image is locally monotonous. The extent of the locality can be chosen before starting the reconstruction. MRP can also include attenuation or scatter corrections and applies as well to transmission reconstruction. However, in contrary to the EM algorithms, MRP does not preserve the total number of measured counts in the image. It does not converge to a ML estimate either. Nevertheless, it shows effective performances in removing the noisy patterns without blurring the locally monotonous structures.

II.7. A Modified OSEM Reconstruction Method based on Partial Volume Effect Correction Techniques (IPVE)

The aim of PET reconstruction based on partial volume effect (PVE) is to estimate the spatial distribution of the isotope concentration on the basis of the projected counts recorded in the scanner.

II.7.1. Partial Volume Effect

As shown on Figure 4, accurate interpretation of brain PET data is mainly obscured by partial volume effects (PVE) due to the inherently limited spatial resolution of PET [Hoffman et al., 1979; Mazziotta et al., 1981]. Quantitative evaluation of positron emitting distribution measured by PET suffers from partial volume averaging of activity across regions of interest. Indeed, in case of cerebral studies, the PET signal reflects the average tracer concentration in each region derived from different cerebral tissue compartments such as grey matter (GM), white matter (WM) and cerebrospinal fluid (CSF). PVE is especially important in the case of cerebral atrophy, which may

lead to an over- or under-estimation of the severity of the disease. As a result of this, it is often impossible to determine the extent to which PET abnormalities represent changes of functional activity in the atrophic structure, structural changes or a combination of the two.

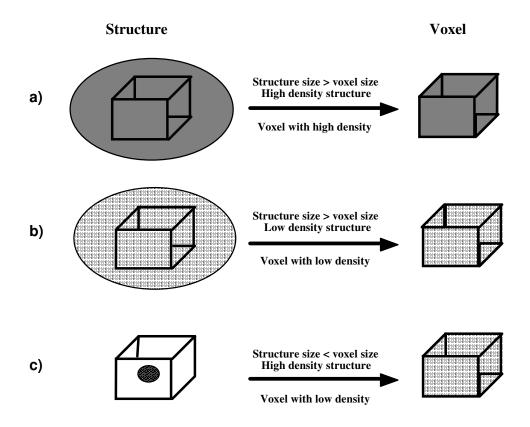


Figure 4: Partial volume effect (spillover) explanation. (a) Case where the structure is of normal uptake density, and of bigger size than the voxel size (i.e high uptake density). The voxel uptake density has the same uptake density than the structure. (b) Case where the structure is of low density, and of bigger size than the voxel size. The voxel uptake density has the same uptake density than the structure (i.e low uptake density). (c) Case where the structure is of high uptake density, and of smaller size than the voxel size. The voxel uptake density has lower uptake density than the structure. The latter reflect the spillover phenomenon also refereed to as partial volume effect (PVE).

II.7.2. Algorithm

Various image post-reconstruction techniques have been developed to incorporate PVE corrections (Mueller-Gartner et al., 1992, Labbé, 1995, Labbé et al., 1998). This sub-section describes a new iterative approach for PVE corrections (IPVE) based on a previously developed method (Labbé et al., 1997) and a modified ML-EM reconstruction. This approach consists in replacing in Eq. 1 the MLEM current image by an *ideal image estimate* simulating very high resolution PET data corrected for PVE. Mapping constraints derived from high resolution anatomical images, like MR or CT images, of the same patient, it attempts to relocate the distribution of PET activity directly on the anatomical image itself.

Let first describe basics of the PVE correction method. Assessment of PVE requires a knowledge of the size and shape of the structures of interest, of their relative position in the scanner gantry, and of the presence of surrounding

high or low activity [Hoffman et al., 1979; Kessler et al., 1984; Mazziotta et al., 1981; MŸller-Gartner et al., 1992]. Using an automated MR segmentation (see section V), we have developed a new method for PVE correction of PET data, which is based on a the linear least square (LLS) minimisation (Labbé et al., 1998). This approach allows simultaneous measurement of an unlimited number of tissue compartments. Segmenting MR into Boolean or probabilistic anatomical maps partitioned in different tissue compartments results in N distinct images of volumes of interest (VOI) I_{VOI_k} (k=1,...,N) denoting the probability maps of tissue VOI_k . Let us now assume that each image I_{VOI_k} is associated with a parameter C. C may represent a concentration of tracer, or a parameter derived from the analysis of PET data. And let PSF be the point spread function of a given PET scanner. Then the PET image I_{PET} may be expressed as:

$$I_{PET} = \left(\sum_{k=1}^{N} C_{VOI_k} I_{VOI_k}\right) \otimes PSF$$
 (Eq. 13)

where the symbol \otimes stands for the convolution product.

Since the convolution product is a linear operation, Eq. 13 can be expressed for each pixel with a set of linear algebraic equations written in a matrix form as:

$$\text{where } b = \begin{bmatrix} PET_1 \\ PET_2 \\ \vdots \\ PET_j \\ \vdots \\ PET_M \end{bmatrix}; \ A = \begin{bmatrix} A_{VOI_{11}} & A_{VOI_{21}} & \cdots & A_{VOI_{k_{1}}} & \cdots & A_{VOI_{N_{1}}} \\ A_{VOI_{12}} & A_{VOI_{22}} & \cdots & A_{VOI_{k_{2}}} & \cdots & A_{VOI_{N_{2}}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{VOI_{1j}} & A_{VOI_{2j}} & \cdots & A_{VOI_{k_{j}}} & \cdots & A_{VOI_{N_{j}}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{VOI_{1M}} & A_{VOI_{2M}} & \cdots & A_{VOI_{k_{M}}} & \cdots & A_{VOI_{N_{M}}} \end{bmatrix}; \ x = \begin{bmatrix} C_{VOI_{1}} \\ C_{VOI_{2}} \\ C_{VOI_{k}} \\ C_{VOI_{N}} \end{bmatrix}.$$

- b is the PET image of size M, where M is the total sum of pixels in the whole image volume (i.e. $M = dimx \times dimy \times Nplane$ where dimx, and dimy, are the x-axis and y-axis image dimensions, respectively, and Nplane is the number of image planes);
- $A = [A_{VOI_1}, A_{VOI_2}, ... A_{VOI_N}]$ is a matrix of $M \times N$ where N is the total number of VOIs, each column of A representing the convolution product of the probabilistic or Boolean segmented MR image within one VOI or tissue compartment (i.e. $A_{VOI} = I_{VOI} \otimes PSF$);
- x represents the cerebral tissue activity concentrations which is the unknown vector to be estimated (i.e. C_{VOI_1} , C_{VOI_2} , ... C_{VOI_N}).

An example of the segmented and convolved anatomical images is shown in Figure 5 for a MR brain study. These MR images have been segmented into probabilistic maps using the SPM segmentation (Ashburner et al., 1997).

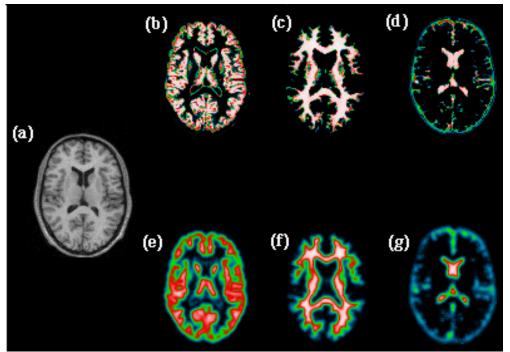


Figure 5: Convolution of probabilistic segmented MR brain image. (a) Original MRI. The segmented probabilistics anatomical maps are shown on top for (b) grey matter (GM), (c) white matter (WM), and (d) cerebral spinal fluid (CSF). Their respective convolution with the 2D point spread function of the ECAT-953 PET scanner are shown on the bottom (e, f, and g).

The best solution of the matrix equation Ax = b is the one which comes closest to satisfying all equations simultaneously. Using the least square method (i.e. minimising the sum of the squares of the differences between the left and right-hand sides of equation Ax = b), the set of linear equations can be reduced to a solvable linear least-squares (LLS) problem. The computation of $x = A^{-1}b$ is then obtained by decomposition of A into singular values [Press et al., 1988].

The application of this PVE correction method using a modified ML-EM reconstruction is implemented as follows:

- 1) Convolve the segmented anatomical images I_{VOI_k} (k=1,...,N) with the 2D PSF of the PET scanner so that the latter is on the same resolution than PET reconstructed images.
- 2) Initialise concentration values C_{VOI_k} (k=1,...,N) to the invert of the number of labelled tissue existing in the segmented anatomical image 1/N.
- 3) Multiply $(I_{VOI_k} \otimes PSF)$ by C_{VOI_k} (k=1,...,N) to obtain an *ideal image* estimate (Eq. 13).
- 4) Incorporate the *ideal image* estimate of step 3 in the ML-EM equation (Eq.1) and compute a new image estimate of the activity distribution.
- Invert Eq. 14 using the new image estimate of step 4 to obtain a new set of concentration values C_{VOI_k} (k=1,...,N).
- 7) Iterate from step 3 until appropriate convergence has been achieved.

In the case of phantom data, either simulated by Monte Carlo or measured on a given scanner, the *ideal image* can be obtained by multiplying each estimated concentration values with the prior image sorted out directly from the segmented emission image. Otherwise, with real patient data, since the segmentation of the emission image might represent a severe problem for a certain studies, it is better to construct an *ideal image* by multiplying each estimated concentration values with the segmented anatomical image convolved by the PSF of the scanner as shown in Eq. 13. This iterative partial volume effect (IPVE) correction method will then result in a high resolution anatomical image displaying PET tracer activity concentrations.

Contrary to the Bayesian approach, here the a priori information is not reflected by an energy function applied on pixel intensities within a given neighbourhood. However, PVE correction methods using OSL-type reconstructions could also be envisaged the same way, but were not investigated in this study.

II.7.3. Conditions

The PVE correction method presented here is being highly susceptible to errors in the requisite registration such as AIR (Woods et al, 1993, Ashburner et al., 1997) and segmentation procedures (Boudraa 1997, Liang et al., 1993, Raya, 1990). Certainly, such localisation methods are dependent on unrealistic simplifying assumptions made on the activity ditributions in the brain or in the abdomen. Nevertheless, a good registration and an accurate segmentation should minimise errors introduced during the PVE correction. Moreover, the PVE correction method can be applied on either Boolean, or even better probabilistic anatomical images. The latter option has the advantage to be more accurate since PVE corrections can be performed into smaller and smaller regions until small homogeneous volumes of interest are obtained.

I	l I				С	0	M	Р	Α	R	ı	S	0	N			0	F
M	Α	Ρ			R	Ε	C	0	N	S	Т	R	U	С	Т	ı	0	N
W	I	T	Н					0	S	Ε	M					Α	N	D
Α	N	Α	L	Υ	Т	ı	С											
R	Ε	C	0	N	S	Т	R	U	С	Т	I	0	N					
Α	L	G	0	R	1	T	Н	M	S									

III.1. Results On Hoffman Brain Phantom Through Reconstructed Images

Using simulated Hoffman brain phantom corrected for both scatter and attenuation obtained with *Eidolon* Monte Carlo simulator (see D2.1), several reconstruction algorithms have been compared for a slice located at the middle of brain phantom. These reconstruction algorithms as implemented in Figure 6 are

- 3D reprojection (or PROMIS) with pure Ramp (fc=.5, alpha=1)
- 3D reprojection (or PROMIS) with Ramp apodized by a Hamming window (fc=.5, alpha=.5)
- Fourier rebinning (FORE) with a Ramp filter apodized with a Hamming window (fc=.5, alpha=.5)
- Ordered Subste-Expectation Maximization (OSEM) with no filter (6 subsets, 18 subiterations)

- Ordered Subste-Expectation Maximization (OSEM) with no iterative Metz power (xdir=1, zdir=1) and (6 subsets, 18 subiterations)
- Median Root Prior -(MRP-OSEM) with 3 subsets, and 9 subiterations and a beta value = 0.001
- Markov Random Field with Green prior (MRF_GR) with 3 subsets, and 9 subiterations and a beta value = 0.001
- Iterative Partial Volume Effect (IPVE) with 3 subsets, and 9 subiterations

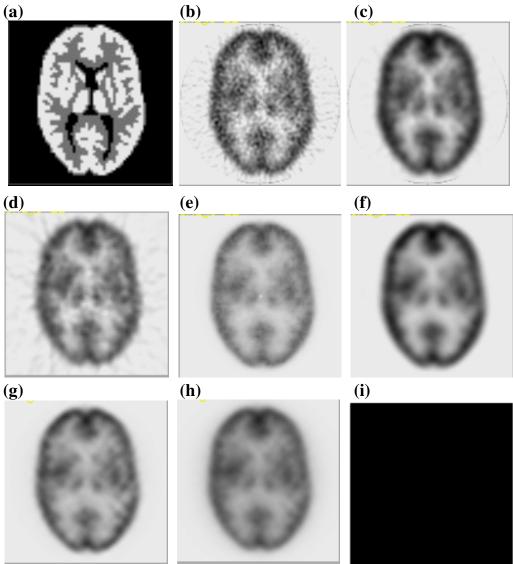


Figure 6: Example on brain Hoffman phantom using simulated projection data from Eidolon Monte Carlo simulator: (a) Reference image (b) PROMIS with pure Ramp filter (fc=.5, alpha=1) (c) PROMIS with ramp apodized with Hamming filter (fc=.5, alpha=.5), (d) FORE (fc=.5, alpha=.5) (e) OSEM with no iterative Metz power, 6 subsets, and 18 subitterations, (f) Same than the previous but OSEM with iterative Metz power (xdir=1, zdir=1), (g) MRP 3subsets, and 9 subitterations and beta=0.001, (ih MRF_GR) with 3 subsets and 9 subitterations, beta=.2 (e) IPVE (6 subsets, and 18 subitterations),

III.2. Results On Utah Phantom Trhough Mediane Profiles

We present here results based on median image profiles of an Utah phantom on a central plane of the scanner. These image profiles are obtained:

- by reprojecting a reference image of Utah phantom containing only three values (2 for the concentric cercle and the hot spot, 1 for the cylinder background, and 0 for the cold spot and background),
- then by implementing the reconstruction algorithim from these forward projected data (projection data).

This has the advantage to evaluate the performance of those algorithms by calculating simple figure of merits (FOMs) (i.e. contrast, coefficient of variation, activity ratio between two regions...) based on the use of regions of interest. Tough we will not present here theses FOMs, but we will just illustrate the profiles of some reconstructions algorithms specially the MAP algorithms.

application of the proposed MAP algorithms by using the priors:

In figure 7, median image profiles are shown by pairs representing on one hand the reference image profile and on the other hand the used reconstructions algorithms profile such as

- 3D reprojection (or PROMIS) with pure Ramp (fc=.5, alpha=1)
- 3D reprojection (or PROMIS) with Ramp apodized by a Hamming window (fc=.5, alpha=.5)
- Fourier rebinning (FORE) with a Ramp filter apodized with a Hamming window (fc=.5, alpha=.5)
- Ordered Subset-Expectation Maximization (OSEM) with no filter (6 subsets, 18 subiterations)
- Median Root Prior -(MRP-OSEM) with 3 subsets, and 9 subiterations and a beta value = 0.001
- Markov Random Field with Green prior (MRF GR)
- Iterative Partial Volume Effect (IPVE) with 3 subsets, and 9 subiterations (based on the multiplication of new estimed values by the segmented and convolved MRI)
- Iterative Partial Volume Effect (IPVE) with 3 subsets, and 9 subiterations (based on the multiplication of new estimed values by the only segmented MRI)

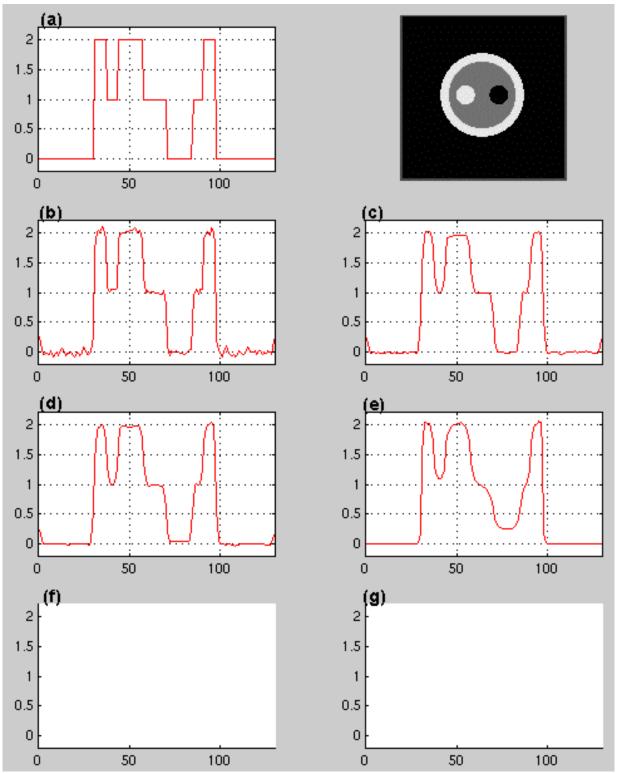


Figure 7: Median image profiles of the Utah phantom on a central plane of the scanner: (a) reference image, with its image, (b) PROMIS with pure ramp (fc=.5, alpha=1), (c) PROMIS with Ramp filter apodized with a Hamming window (fc=.5, alpha=.5), (d) FORE (fc=.5, alpha=.5), (e) OSEM with no filter (6 substes, 18 subiteration), and (bf MRP with beta=0.001 (3 subsets, 9 subiteration), (g) MRF_GR with beta=0.001 (3 subsets, 9 subiteration), where the ideal PET image has been obtained the new estimated values have been multiplied by the segmented and convolved MR images, h) Same as previous one but the ideal PET image has been the new estimated values have been multiplied by the only segmented MR images.

I V . M A P B U I L D I N G B L O C K S

IV.1. MAP Method Class

Description:

This class contains the image prior information used in MAP reconstruction algorithm.

Location:

This class is declared in *include/MAPOSEM/PETMAPParameters.h*.

Class:

Enumeration:

• enum MAPPotentialFunction{GM=0, GR, HL, HS, HU, PS, None};

This is the set of potential functions used for Markov Random Field algorithms with:

- GM: German and Clure method;
- GR : Green method;
- HL: Hebert and Leahy method;
- HS: Hyperspace.
- enum MAPMethod{ MRF_GM=0, MRF_GR, MRF_HL, MRF_HS, MRP, PVE,GAUSS,SGAUSS,NOMAP};

This is the set of prior image used in MAP reconstruction algorithm with:

- MRF: Markov Random Fields;
- MRP: Median Root Prior;
- GRF: Gibbs Random Fields;
- PVE: High resolution image prior based on partial volume effect correction;

- GAUSS : Gaussian prior;
- SGAUSS: Smoothness-Gaussian prior.

Constructors:

• MAPModel(MAPMethod MAPmodel_v = MRP)

This constructor assigns the a priori model to MAPmodel instance

• MAPModel(MAPPotentialFunction Vf_v = GR)

This constructor assigns the a priori model to Vf instance.

• MAPModel(MAPMethod MAPmodel_v, MAPPotentialFunction Vf_v)

This constructor assigns both the a priori model and the potential function to MAPmodel and Vf instances.

IV.2. MAP Reconstruction Parameters Class

Description:

This class contains the MAP parameter informations used in MAP reconstruction algorithm and all the members functions.

Location:

This class is declared in *include/MAPOSEM/PETMAPParameters.h* and *iterative/MAPOSEM/PETMAPParameters.cxx*.

Class:

/* General Reconstruction Parameters (see parameters details in draft 4.1)*/

```
const string input_filename_v = "",
const string output_filename_prefix_v = "",
const int new_size_v=-1,
const double zoom_v=1.,
```

```
const double Xoffset_v=0.,
const double Yoffset_v=0.,
const int max_segment_num_to_process=-1,
const int num_views_to_add_v=1,
const int disp_v=0,
const int save_intermediate_files_v=1,
```

/* OSEM Parameters (see parameters details in draft 5.2) */

```
const int MAP_mode = 0,
const double delta=1.0,
const double kappa=0.01,
const int num_subsets = 1,
const int phase_offset = 0,
const int num_subiterations = 1,
const bool zero_seq0_end_planes = 0,
const string initial_image_filename = "1",
const string sensitivity_image_filename = "1",
const int save_interval = 1,
const int iterative_filter_type = 0,
const int iterative_filter_interval = 1,
const double iterative_filter_fwhmx_dir = .5,
const double iterative_filter_fwhmz_dir = .5,
const double iterative_filter_Nx_dir = 0.0,
const double iterative_filter_Nz_dir = 0.0,
const int post_filter_type = 0,
const double post_filter_fwhmx_dir = .5,
const double post_filter_fwhmz_dir = .5,
const double post_filter_Nx_dir = 0.0,
const double post_filter_Nz_dir = 0.0,
const bool randomise_subset_order_v = false,
const string additive_sinogram_filename = "0",
```

/* MAP Parameters */

```
const string prior_filename="",
const MAPMethod MAPmodel=MRP,
const MAPPotentialFunction Vf= UNKNOWN,
const double beta = 0.001);
```

This constructor takes all the detailed input parameters in order to derive each parameter from PETReconstructionParameters, PETOSEMParameters, so that PETMAPParameters can inherit these parameters.

```
    virtual string parameter_info();
```

This method returns the MAP reconstruction parameter class.

```
char* MAPmodel_info();
```

This method returns into a string the MAP a priori model.

```
• char* PotentialFunction_info();
```

This method returns tinto a string the MAP potential function.

IV.3. MAP Reconstruction Class

Description:

This class contains the MAP reconstruction algorithm and their members functions.

Location:

This class is declared in *include/MAPOSEM/PETMAPReconstruction.h* and *iterative/MAPOSEM/PETMAPReconstruction.cxx*.

Class:

```
class PETMAPReconstruction: public PETReconstruction, public PETMAPParameters
public:
   • PETMAPReconstruction(
       string input_filename_v = "",
       string output_filename_prefix_v = "",
       int new_size_v=-1,
      double zoom_v=1.,
      double Xoffset_v=0.,
       double Yoffset_v=0.,
       int max_segment_num_to_process_v=-1,
      int num_views_to_add_v=1,
      int process_by_view_v = 1,
       int disp_v=0,
      int save_intermediate_files_v=1,
/* MAP Parameters */
       string prior_filename_v="",
       MAPMethod MAPmodel_v=MRP,
       MAPPotentialFunction Vf_v= UNKNOWN,
       double beta_v=0.001,
/* OSEM Parameters */
      int MAP_mode_v=0,
      double delta_v=1.0,
      double kappa_v=0.01,
       int num_subsets_v=1,
       int phase_offset_v=0,
       int num_subiterations_v=1,
      bool zero_seg0_end_planes_v=0,
       string initial_image_filename_v="1",
       string sensitivity_image_filename_v="1",
       int save_interval_v=1,
       int iterative_filter_type_v=0,
       int iterative_filter_interval_v=1,
       double iterative_filter_fwhmx_dir_v=.5,
       double iterative_filter_fwhmz_dir_v=.5,
       double iterative_filter_Nx_dir_v=0.0,
       double iterative_filter_Nz_dir_v=0.0,
       int post_filter_type_v=0,
       double post_filter_fwhmx_dir_v=.5,
       double post_filter_fwhmz_dir_v=.5,
       double post_filter_Nx_dir_v=0.0,
       double post_filter_Nz_dir_v=0.0,
       bool randomise_subset_order_v=false,
       string additive_sinogram_filename_v="0");
```

This constructor takes as input parameters, the list of detailed parameters needed for the MAP reconstruction:

- the common parameters being used for filtering (fsaving, display, maximum segment number to be processed);
- the parameters being used for MAP.

```
• PETMAPReconstruction(const PETMAPParameters &parameters);
```

This constructor takes as input parameters, the list of parameters needed for the MAP reconstruction

Data access method:

// Method for reconstruction

```
    virtual void reconstruct( PETSinogramOfVolume &s3d,
    PETImageOfVolume &image,
    PETImageOfVolume &PriorImage);
```

This method implements the reconstruction by giving as input the emission sinogram data corrected for attenuation, scatter, dead time (see detail in D3.2 section Correction Info in emission, transmission...), and returns the output reconstructed image.

```
    void reconstructPVE( PETSinogramOfVolume &s3d,
    PETImageOfVolume &image,
    PETImageOfVolume &PriorImage);
```

// Methods for the parameters aand MAP model informations

```
virtual string parameter_info();
```

This method returns all the parameters used for the MAP reconstruction algorithm as described just above in constructor section.

```
virtual string method_info()
```

This method returns the type of the a priori model during the MAP reconstruction.

// MAP utilities

This method updates the current image obtained by ML-EM and incorporates the a priori information through one the MAP a prior model.

// MAP reconstructions

```
    void PETMAP_MRF_GM( const PETImageOfVolume &ImagePrior,
    PETImageOfVolume &Derivative_Vc);
```

This method returns the recontructed image *lambda* based on Markov Random Field using Gibbs priors based on Geman and McClure method.

```
    void PETMAP_MRF_HL( const PETImageOfVolume &ImagePrior,
    PETImageOfVolume &Derivative_Vc);
```

This method returns the recontructed image *lambda* based on Markov Random Field using Gibbs priors based on Hebert and Leahy method.

```
    void PETMAP_MRF_HS( const PETImageOfVolume &ImagePrior,
    PETImageOfVolume &Derivative_Vc);
```

This method returns the recontructed image *lambda* based on Markov Random Field using Gibbs priors based on hyperspace method.

```
    void PETMAP_MRP( const PETImageOfVolume &ImagePrior,
    PETImageOfVolume &Derivative_Vc);
```

This method returns the recontructed image lambda based on Median Root Prior.

```
    void PETMAP_GAUSS( const PETImageOfVolume &ImagePrior,
PETImageOfVolume &Derivative_Vc,
int smooth);
```

This method returns the recontructed image *lambda* based on Gaussian prior.

```
    void PETMAP_SGAUSS( const PETImageOfVolume &ImagePrior,
    PETImageOfVolume &Derivative_Vc,
    int smooth);
```

This method returns the recontructed image lambda based on smoothness Gaussian prior.

```
    void PETCreatePSFGauss( PETPlane &PSFGauss, double FWHM);
```

This method creates the 2D point spread function (PSF) image representing the resolution of studied PET scanner which is expressed as Full Width at Half Maximum (FWHM).

```
• void PETMAP_PVE( PETImageOfVolume &lambda, PETImageOfVolume &ImageAnatomical);
```

This method returns the recontructed image *lambda* based on high resolution PET prior corrected for partial volume effect.

// MAP reconstruction utilities (partial volume effect correction)

```
• Tensor4D<float> PETMAP_ConvolvePrior(PETImageOfVolume &ImageAnatomical);
```

This method takes as input the prior anatomical image (*ImageAnatomical*) in order to convolve each tissue compartment (labelled in *ImageAnatomical* image) with the 2D point spread function (PSF) of the PET scanner, and returns therefore a 4D tensor representing the *N* tissue compartments which are include in the 4th dimension, convolved with the 2D PSF.

This method computes and returns a simulated PET image (*lambda*) obtained by multiplying each tissue estimated concentration (estimated_concentration) with only the segmented anatomical image of this tissue (*ImageAnatomicalTissue*).

This method computes and returns a simulated PET image (*lambda*) obtained by multiplying each tissue estimated concentration (estimated_concentration) with the segmented and convolved anatomical image of this tissue, where the tissue compartment number is obtained from the 4th dimension of *ImageAnatomicalTissue_convlv*. This returns an high resolution PET image.

This method computes and returns as output the estimated concentration vector from the current image *lambda* and the segmented and convolved anatomical image (i.e. the prior image or *ImageAnatomicalTissueConvolved*).

```
void do_log_file();
```

This method prints out the log file (i.e. the MAP reconstruction parameters, CPU and true time) for the MAP implementation.

Other external functions:

```
    void PETMAP_SingValDecomp( Tensor2D<float> &a,
    Tensor1D<float> &w,
    Tensor2D<float> &v);
```

This function computes the singular value decomposition, $A = U W V^T$, given a matrix A[1..m][1..n]. The matrix U replaces A on output. The diagonal matrix of singular values W corresponds to output vectors W[1..n]. The matrix V (not the transpose V^T) is output as V[1..n][1..n].

This function solves Ax = b where A has been computed through the previous function $A = U W V^T$, and x is the outure solution vector.

```
    void QuickSort(Tensor1D<float> &array, int min_index, int max_index);
```

This function sorts a partial array based on the partition of the elements in the subarray (min_index..max_index) so that the pivot value is in place in order to quicksort for each subarray.

```
    void build_subset_array( int *subset_array, int array_length);
```

This function generates a new subset sequence (See more details in D5.2).

This function computes the penalty derivative (See more details in D5.2).

V. SEGMENTATION

The segmentation process is based on two different methods: the first one is user's dependent in opposition to the second one that is automated.

V.1. Segmentation by User's Defined Thresholds

This method involves a preliminary knowledge of the grey level's distribution in the image to be segmented. The user defines the thresholds (grey level values) in order to separate the different features in the image. The segmentation is performed by the assignment of a different label to each feature.

V.2. Automated Segmentation by Fuzzy C-Means Method

The aim of this method is to automatically partition data into subsets (or clusters). This partition is based on the similarity between elements within the same cluster.

The particularity of the Fuzzy C-means algorithm (FCM) is that every pixel is characterised by degrees of belonging to each cluster. This allows a segmentation that takes into account the fuzziness of the pixel information (Fig. 8). This algorithm is detailed in Boudraa et al. (1997).

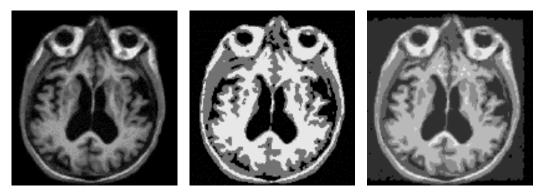


Figure 8: Segmentation by FCM method: (a) original MRI image, (b) result of the segmentation of (a) into 3 clusters, (c) result of the segmentation of (a) into 6 clusters.

V.2.2. Method

The FCM algorithm uses iterative optimisation to approximate minima of an objective function, Jm. Consider a finite set of data points, $X = x_1, x_2, ..., x_n$:

$$Jm(U,V;X) = \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{ik})^{m} \cdot (d_{ik})^{2}$$
 (Eq. 15)

$$V = (v_1, v_2, ..., v_c)$$
 (Eq. 16)

subject to $\mu_{ik} \in [0,1]$, $U = \sum_{i=1}^{c} \mu_{ik} = 1$ with $1 \le k \le n$, and $\sum_{k=1}^{n} \mu_{ik} \in]0,n[$ with $1 \le i \le c$ where:

- n: number of data to be clustered;
- c: number of clusters;
- m (>1): scalar measuring the fuzzification degree,
- μ_{ik} : membership degree of k to cluster i,
- v_i : centre of cluster i,
- d_{ik} : measure of similarity between v_i and x_i (pixel intensity).

V.2.3. FCM algorithm

This algorithm is used in the segmentation class.

(a) Choose *c* clusters, with initial membership values:

$$\sum_{i=1}^{c} \mu_{ik} = 1$$
 (Eq. 17)

(b). Calculate the c clusters centre:

$$v_{i} = \frac{\sum_{k=1}^{n} (\mu_{ik})^{m} \cdot x^{k}}{\sum_{k=1}^{n} (\mu_{ik})^{m}}$$
(Eq. 18)

(c) Compute new membership values:

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c} \left(d_{ik} / d_{jk} \right)^{(m-1)}}$$
 (Eq.19)

Return to point (b).

(d) Compute the defect measure:

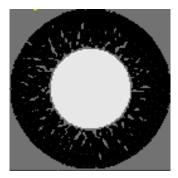
$$E = \sum_{i=1}^{c} \left| \tilde{\nu}_i - \nu_i \right| \tag{Eq. 20}$$

While (E>threshold value), return to points (b) and (c).

V.3. Processing of The Image Before Segmentation

A choice is given to the user to apply a filter on the image before running the automatic segmentation (FCM). The purpose is to remove noise by modifying the value of each pixel in function of the neighbouring pixels included in a 3 dimensional kernel. Two methods are proposed:

- *median filter*: the central pixel takes the value of the median inside the kernel. This method is edge preserving but involves a huge computational time: 125 operations for each pixel with a cubic kernel-size of 5.
- *smoothing*: the central pixel takes the value of the average of all pixels inside the window. This one doesn't preserves edges. This operation can be performed independently along the three axis in a 1D way, leading to the same result than in a 3D way. This involves a sensible gain in the computational time.



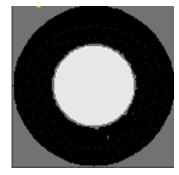


Figure 9: Simulated PET cylinder data: (a) segmentation without the application of a median filter, and (b) segmentation with the application of a median filter (kernel-size of 3)

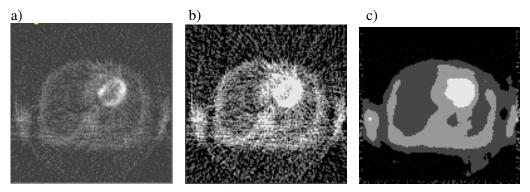


Figure 10: Real oncology PET data: (a) original image, (b) FCM segmentation (4 clusters) without filtering, and (c) same as (b) with filtering (kernel-size of 7).

V.4. Class of Segmentation Process

Description

The class Segmentation contains the implementation of the segmentation process (either manual or automatic).

Location

This class is declared in include/segmentation.h.

Class

```
class Segmentation
public:
     int min_z, min_y, min_x;
                                              // Minimum of the 3-axes configuration
                                               // Minimum of the 3-axes configuration
     int max_z, max_y, max_x;
      float val_inf, val_sup, inter_val; // pixel values in the input image
     unsigned short num_cluster;
                                              // number of clusters
      Tensor3D<short> image_short;
                                              // output image data
private:
                                              // threshold values
      Tensor1D<float> thr_val;
                                              // cluster centers
      Tensor1D<double> cluster_center;
      Tensor1D<double> dist_diff;
                                              // defect measure
```

```
Tensor2D<double> dea:
                                                // membership degrees
      Tensor2D<double> new_deg;
      Tensor1D<long> hist;
                                                // histogram of the input image
      Tensor1D<long> hist_val;
                                                // histogram of the non 0 values
      Tensor1D<unsigned short>.0 label; // labels corresponding to each values
      Tensor1D<unsigned short> order_label;
                                                // idem, but ordered; lowest label
                                                   corresponds to lowest values
public:
// constructor
      Segmentation(const PETImageOfVolume &image,unsigned short num_cluster);
// redistribution of the input values (either integer or float) into integers (0,255)
      void convert(PETImageOfVolume &image);
// application of a median filter before automatic segmentation (optional)
      void median_filter(Tensor3D<short> &image_short, short kernel_size);
// application of an average filter before automatic segmentation (optional)
      void smooth1D(Tensor3D<short> &image_short,short kernel_size);
// compute histogram of converted values
      void histogram(Tensor3D<short> &image_short);
// initial membership values (based on hist_val), FCM step 1
      void membership_degrees();
// clusterization, FCM steps 2,3,4
      void clusterized();
// first labelisation (based on maximum membership degrees)
      void stabilisation();
// ordered labelisation
      void make_label(Tensor3D<short> &image_short);
// manual method; user's defined thresholds
      PETImageOfVolume input_thresh(PETImageOfVolume &image, unsigned short num_cluster);
};
```

The FCM method uses as inputs the original data (either integer or float), and the number of clusters. The data is converted and redistributed between 0 and 255. The choice is given to the user of the application of a filter (method and kernel-size). Then the following member functions are used: histogram, membership_degrees, clusterized, and stabilisation. The output data is of type short where the labels are ordered.

The other method uses as input, in addition to the original data and the cluster number, the threshold values. The only member function used is input_thresh. The output data is from the same type as the original data.

Function

```
static int sortfun ( short *a, short *b);
```

This function is used by qsort() in median_filter() to sort two values.

VI. REFERENCES

Ashburner J. and Friston K.J. (1997) Multimodal Image Coregistration and Partitioning - a Unified Framework.

NeuroImage, Vol. 6, Issue 3: 209-217.

- Alenius S. and Ruotsalainen U. (1997) Bayesian image reconstruction for emission tomography based on median root prior. *European Journal of Nuclear Medicine*, vol. 24 No. 3: 258-265.
- Ardekani B.A., Braun M., Hutton B.F., Kanno I. and Iida H. (1996) Minimum cross-entropy reconstruction of pet images using prior anatomical information. *Phys. Med. Biol.*, 41: 2497-2517.
- Boudraa A.E.O. (1997) Automated detection of the left ventricular region in magnetic resonance images by Fuzzy C-Means model. *Int J of Cardiac Imag.*, 13: 347-355.
- Bouman C.A. and Sauer K. (1993) A generalized gaussian image model or edge-preserving map estimation. *IEEE Trans. Image Processing*, 2: 296-310.
- Geman S. and McClure D. (1985) Bayesian image analysis: an application to single photon emission tomography. in Proc. American Statistical Society, Statistical Computing Section (Washington, DC) 12-18.
- Green P.J. (1990) bayesian reconstruction from emission tomography data using a modified EM algorithm. IEEE Trans.

 Med. Imag. MI-9: 84-93.
- Hoffman E.J., Huang S.C. and Phelps M.E. (1979) Quantitation in positron emission computed tomography: 1. Effect of object size. *J Comput Assist Tomogr.*, 3: 299-308.
- Johnson V.E. et al. (1991) Image restoration using Gibbs priors: boundary modeling, treatment of bluring and selection of hyperparameter. *IEEE Trans. PAMI*, 13: 413-425.
- Kessler R.M., Ellis J.R. and Edden M. (1984) Analysis of emission tomographic scan data: Limitations imposed by resolution and background. *J Comput Assist Tomogr.*, 8: 514-522.
- Kontaxis G., Strauss L.G. and Kaick G.V. (19
- Labbé C., Koepp M., Ashburner J., Spinks T., Richardson M., Duncan J., and Cunningham V. (1998) PET Quantification with Correction for Partial Volume Effects within Cerebral Structures in Quantitative Functional Brain Imaging with Positron Emission Tomography. By Richard E. Carson, Margaret E. Daube-Witherspoon, and Peter Herscovitch, Academic Press., 9,: 59-66.
- Labbé C., Ashburner J., Koepp M., Spinks T., Richardson M. and Cunningham V. (1997) Accurate PET quantification using correction for partial volume effects within cerebral structures. *Neuroimage*, 5: B12.

- Labbé C. (Nov. 1995) Cerebral atrophy measurements from Magnetic Resonance Imaging and correction for metabolic data from Positron Emission Tomography: Application to Alzheimer's disease. *Ph.D. thesis in Medical Imaging (in french), University of Paris XI*.
- Lalush D. and Tsui B. (1995) A fast and stable maximum a posteriori reconstruction algorithm. *Med. Phys.*, 22(8): 1273-1284.
- Lange K. (1990) Convergence of EM Image reconstruction algorithms with Gibbs smoothing. IEEE Trans. Med. Imag., MI-9 4.
- Liang Z. (1993) Tissue classification and segmentation of MR images. IEEE Eng. Med. Biol.: March: 81-85.
- Hebert T. J. and Leahy R. M. (1989) A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors, *IEEE Trans.Med.Im.*, vol. 8, pp. 194-202.
- Mazziotta J.C., Phelps M.E., Plummer D. and Kuhl D.E. (1981) Quantitation in positron emission tomography: 5. Physical-anatomical effects. *J Comput Assist Tomogr.*, 3: 299-308.
- MŸller-Gartner H.W., Links J.M., Prince J.L., Bryan R.N., McVeigh E., Leal J.P., Davatzikos C. and Frost J.J. (1992)

 Measurement of radiotracer concentration in brain gray matter using positron emission tomography: MRI-based correction for partial volume effects. *J Cereb Blood Flow Metab.*, 12: 571-583.
- Press W.H., Flannery B.P., Teukolsky S.A. and Vetterling W.T. (1988) Modeling of data: Non linear models.

 Numerical Recipes: The art of scientific computing, Cambridge, New-York: 521-52.
- Raya S.P. (1990) Low-level segmentation of 3-D magnetic resonance brain images: A rule base system. *IEEE Trans. Med. Imag.*, 9: 3: 327-337.
- Thierry Raphael (Oct. 1999) Correction de l'attenuation de la réponse géometrique d'un détcteur en tomographie par émission sur un colis dee déchets nucléaires. *PhD Thesis (in french), Université Joseph Fourier, Grenoble.*
- Woods R.P., Mazziotta J.C. and Cherry S.R. (1993) MRI-PET registration with automated algorithm. *J. Comput. Assist. Tomogr.*, 17: 4: 536-546.