

# Enhancing Atlas Based Segmentation with Multi-Class Linear Classifiers

Michaël Sdika

Université de Lyon, CREATIS ; CNRS UMR 5220 ; Inserm U1044 ; INSA-Lyon ; Université Lyon 1, France.

*a. Purpose* To present a method to enrich atlases for atlas based segmentation. Such enriched atlases can then be used as a single atlas or within a multi-atlas framework.

*b. Methods* In this paper, machine learning techniques have been used to enhance the atlas-based segmentation approach. The enhanced atlas defined in this work is a pair composed of a gray level image alongside an image of multi-class classifiers with one classifier per voxel. Each classifier embeds local information from the whole training dataset that allows for the correction of some systematic errors in the segmentation, and accounts for the possible local registration errors. We also propose to use these images of classifiers within a multi-atlas framework: results produced by a set of such local classifier atlases can be combined using a label fusion method.

*c. Results* Experiments have been made on the *in vivo* images of the IBSR dataset and a comparison has been made with several state-of-the-art methods such as FreeSurfer and the multi-atlas non-local patch based method of Coupé or Rousseau. These experiments show that our method is competitive with state-of-the-art methods while having a low computational cost. Further enhancement has also been obtained with a multi-atlas version of our method. It is also shown that, in this case, nonlocal fusion is unnecessary. The multi-atlas fusion can therefore be done efficiently.

*d. Conclusions* The single atlas version has similar quality as state-of-the-arts multi-atlas methods but with the computational cost of a naive single atlas segmentation. The multi-atlas version offers a improvement in quality and can be done efficiently without a nonlocal strategy.

## I. INTRODUCTION

Atlas based segmentation [1–3], is now a common image processing tool. Using an image registration procedure, the geometric transform from a subject image to an image with known segmentation, an atlas, is estimated and used to map the atlas segmentation to the subject image. The only constraint on the atlas is to be similar to the subject image up to a geometric (possibly nonlinear) geometric transform. Atlas based segmentation has been used extensively on human brain data [4, 5] but it has also successfully been used on large variety of medical image segmentation problems such as bee brain [6], heart [7, 8] or prostate [9], for example. In practice, atlas based segmentation is used to measure the size or the atrophy of specific anatomical structures. For example in [10], it allows to investigate clinical factors influencing the size hippocampus. Hippocampus atrophy is indeed an indicator of dementia and cognitive decline. It can also be used to delineate structures for treatment planning in radiotherapy either when the tumor does not deform the anatomy [11–13] or even when the tumor does deform the anatomy [14, 15]. Atlas based segmentation also allows measuring physiological parameters in specific structures. It has been used for example to measure volumetric bone mineral density on quantitative computed tomography images [16], or to quantify the blood flow in the vessel on cardiac images [17].

The segmentation obtained from one atlas depends on the registration quality. But registration is an ill posed problem and the choice of the similarity metric, the transformation model and the degree of regularization that would produce accurate and robust results is difficult. Furthermore, as it is a non-convex nonlinear problem, local minima must be avoided during its numerical reso-

lution.

To improve the segmentation results, a set of several atlases can be used instead of a single atlas. For example, STAPLE was introduced in [18]; this method produces an agreement among a set of manual binary segmentation volumes. In [19], this method has been extended to the problem of multi-label segmentation using multiple atlases. In [20], it is proposed to use shape based averaging to fuse multiple atlas segmentation maps: distance maps to each label and each atlas output is computed and the final labeling minimizes the mean distance map of each label. Majority voting has also been investigated in [6, 21, 22] to fuse the results of multiple atlas segmentation. This method, while simple to implement, does clearly improve the segmentation with respect to the best segmentation of all atlases. In [7, 23], the vote is weighted by a function of the local registration residual. The work of [24] formalizes these kinds of fusion methods using a probabilistic framework. This framework is extended in [25] to include the joint registration of all the atlases to the subject and support intermodality. In [5], the accuracy map of an atlas is introduced and proposed as an original weighting for a vote. A mapping consistent with tissue classification is also proposed. The combination of both propositions provides a clear improvement. The work of Wang [26] is also based on a weighted vote with offline computed weights. The novelty here is to account for the correlation between the atlases for the estimation of the weights. In [9], a weighted vote procedure is also proposed and the weights are estimated iteratively: the segmentation maps from each atlas are fused using a weighted vote, then the weights are re-estimated by comparing each atlas segmentation to the fused segmentation. In [27], the weights are also computed iteratively but using the Expectation Maximization procedure. To

remove the cost of the registration to each atlas of the registration, it has been proposed in [28] to create an average atlas from the atlas dataset and to keep the transformation from the average atlas to each individual atlas. To segment a new subject, only the registration to the template is required, the transformation from the subject to each atlas is obtained by composition.

The fusion methods proposed in [4, 29] are both inspired by the non-local mean denoising method presented in [30]. For these methods, the fusion is done using a vote in which the matching voxel in each atlas and their neighbors contribute. The difference between the local patch around the subject voxel and the local patch around each contributing voxel of the atlases is used to weight the vote. This non-local fusion method has also successfully been used on computed tomography (CT) images for the segmentation of the prostate in [31] or for the parathyroid gland in [32]. The feature used in these cases was local patches or HOG features [33].

The improvement offered by multi-atlas segmentation is really important, especially when nonlocal fusion is used. However, this benefit comes with a higher computational cost: several registrations need to be done and whereas the multi-atlas fusion cost itself is often negligible, it is not when nonlocal fusion is used.

A different approach to improve atlas based segmentation is to gather information complementary to the label map (such as local statistics) into the atlas. In FreeSurfer [34], the atlas has been enriched with local intensity distributions and label neighborhood statistics. Machine learning has been used for medical image segmentation in [35–37]. In these works, a classifier is trained for each structure to learn the decision functions that will segment the structures. The local features used for the training are augmented with the coordinates of the voxels for a position dependent response of the classifier. A Bayes classifier with Markov prior is used in [38] to segment a single structure using multi-scale Gaussian derivatives features. A probabilistic atlas is used to provide prior on the structure location. In the atlas based segmentation context, machine learning can be used to enhance an atlas: in the training step, a decision function is optimized for the task of segmenting images based on known samples. Once the model has been learned offline, it can hopefully be efficiently used for the online segmentation of a new subject.

In the work of [39], a random forest  $a_i$  is trained on each atlas of the dataset. Each tree of  $a_i$  is a predictor for one label of the segmentation. The features used are a fourth dimensional vector containing local intensity statistics. To segment a new subject, each voxel of the subject image should be processed with each tree, of each atlas forest, of each atlas of the dataset. The work presented in [37] propose to train a classifier using local patches as features on the misclassified voxel of any given segmentation method to correct its systematic error. The procedure acts as a post processing using one classifier per segmentation label.

Only very recently, some methods have been proposed where local classifiers are trained on the image to help the segmentation. In [40], a linear classifier is trained for each voxels from a set of atlases using 400 dimensional feature vectors. The method is used for binary segmentation and still requires the registration to the whole dataset for an atlas selection step. In [41] local non-linear SVM classifiers are trained on the atlas. Despite the fact that it is used for 2D images, the classifiers are trained with only a subset of the pixels to save training time. Regarding the segmentation time, their SVM based method is more computationally expensive than their non-local inline fusion method. Local classifiers are also used in [42] for the segmentation of 2D ultrasound cardiac images: local random forests are trained for each pixel with a local intensity patch as feature vector.

In the present work, we use machine learning to introduce additional information to the mapping process. In the training phase, using a set of atlases, local linear SVM classifiers learn a new mapping from local features with the goal to correct segmentation error due to registration failures. To segment a new subject, the method can be used either with the registration to a single trained atlas and can be further improved if the registration to several trained atlases is done. Indeed, our segmentation time is equal to a standard single atlas segmentation while still having a quality similar to the multi-atlas patch based method proposed by Coupé or Rousseau [4, 29]. While presenting some similarity with the approach presented in [40–42], our method seems to have an advantage regarding the inline segmentation time. Indeed, while they report an important overhead for the application of their classifiers, the single atlas method presented here has no noticeable overhead over standard single atlas segmentation thanks to the use of linear classifiers, a compact feature vector and a careful implementation: our local SVM are evaluated only when necessary. This efficiency allows us to make a second contribution by including local classifier atlases in a multi-atlas framework: if a set of segmentation maps is generated by a set of classifier images, they can be combined with any multi-atlas fusion method to enhance the results. As we will see, costly non-local fusion methods are unnecessary. Local classifier atlases seem very good at capturing neighborhood information and this allows an efficient multi-atlas fusion.

The method has been evaluated with experiments on *in vivo* Magnetic Resonance (MR) images of the brain and compared to the state-of-the-art methods FreeSurfer and nonlocal multi-atlas fusion [4, 29].

## II. THEORY

### A. Standard Atlas Based Segmentation

For the standard atlas-based segmentation, an atlas is a couple composed of a gray level image  $I_a$  and the corresponding segmentation map  $S_a$ . To segment a gray level

image  $I$  using this atlas, the geometric transform  $T$  that maps the subject image voxels in the atlas coordinate space has first to be estimated. Typically,  $T$  is the output of any image registration algorithm. To compute the segmentation of  $I$ , the atlas segmentation is then mapped through the  $T$  transform:

$$S(x) = S_a(T(x)). \quad (1)$$

The segmentation clearly depends on the quality of the image registration and any local misregistration will directly affect the segmentation. Unfortunately, these local misregistrations are frequent due to the very large number of local minima of the nonlinear registration problem.

### B. Atlas Based Segmentation with Machine Learning

Image registration is a difficult problem: it is ill-posed and the presence of a huge number of local minima complicates its resolution when the images are complex. This induces a non-negligible uncertainty in its output that one needs to take into account for the segmentation. From now on, it is no longer assumed that the transformation given by the registration maps each subject image voxel to its true corresponding voxel in the atlas. It is only assumed that a voxel is mapped in a neighborhood of its true matching position.

The problem is now to discriminate between the voxels in this neighborhood to choose the correct label. For such a discrimination between voxels to be possible, some new information is needed. In our framework, it takes the form of a feature vector that is returned by a function  $F$ : for a given voxel  $x$  of a given image  $I$ ,  $F$  returns a  $d_F$ -dimensional local feature vector:  $F(I, x) \in \mathbb{R}^{d_F}$ . The features can be of any type (shape, texture, contrast, etc.) and should be complementary to the (imperfect) localization information provided by the atlas registration. Several local descriptors have been proposed in the literature based on the estimation of the signal intensity, its derivative or local histograms for example (for example HOG [33], SIFT [43], SURF [44], or LBP [45]). The feature vector can be chosen depending on the application. Indeed, specificities of the modality and the organ one wants to segment can be taken into account in the design of the feature vector. If the label can be estimated given the feature vector, systematic errors of the registration can be reduced. This finally comes down to a multi-class, supervised classification problem: for each neighborhood in the atlas, we need a multi-class classifier that, given a feature vector, returns a segmentation label. An atlas is now considered differently: as a pair composed of the gray level image  $I_a$  and an image of multi-class classifiers  $C_a$ . Each voxel of  $C_a$  is a multi-class classifier, that, given a feature vector, will return a label.

To segment a new image  $I$ , the transformation  $T$  from an image  $I$  to the atlas is first computed from any im-

age registration procedure. Then the segmentation label of the voxel  $x$  is found by applying the corresponding classifier in the atlas,  $C_a(T(x))$ , to the feature vector computed on the subject image at  $x$ ,  $F(I, x)$ . Formally, we obtain:

$$S(x) = C_a(T(x))(F(I, x)). \quad (2)$$

This is what we call MLMapping: instead of the atlas segmentation, this is the output of local classifiers that is mapped through the geometric transform between the subject and the atlas.

### C. Atlas Training

To train the classifier image of an atlas, a dataset composed of  $N_a$  grey level images  $(I_i)_i$  along with their segmentation images  $(S_i)_i$  is needed. Let  $I_a$  be the image on which one wants to train the classifier image  $C_a$ . There is no conceptual restriction on the choice of  $I_a$ . It can, for example, be one of the image of the dataset, it can also be taken as the average of the dataset images or a minimal deformation template [46, 47].

Let  $T_i$  be the invertible transformation from the coordinate space of  $I_i$  to the one of  $I_a$  computed by a chosen image registration procedure and  $T_i^{-1}$  its inverse. To account for the uncertainties due to the potential errors of the registration, it is only assumed that, for a voxel  $x$  in  $I_a$ , its true corresponding voxel in  $I_i$  is in  $V_i(x)$ , the box of size  $N_s \times N_s \times N_s$  centered on  $T_i^{-1}(x)$ .

Thus, for each voxel  $x$ , the classifier in  $C_a(x)$  is trained with the following dataset of  $N_a N_s^3$  feature vectors:

$$\{(S_i(y), F(I_i, y)) \mid i \in [1, N_a], y \in V_i(x)\}.$$

Note that when the registration is symmetric,  $T_i^{-1}$  can be computed directly by switching  $I_a$  and  $I_i$  in the registration routine. In the general case, it is preferable to perform the registration during the training as it will be during the segmentation: with the atlas as the moving image and the subject as the fixed one.

### D. Further Improvement Using a Multi Atlas Combination Method

So far, only one registration is required for the segmentation of a new subject. Information from the whole atlas dataset is gathered on a classifier image. To segment a new subject, only one registration, to the classifier image space, needs to be computed while still using the information from the whole dataset.

This is interesting from a computation point of view but the segmentation quality can be further improved with a multi-atlas version of our method. Consequently, if the training dataset contains  $N_a$  images, each image can be trained as a classifier atlas using the method described in the previous section. For the segmentation of a

new subject, its image should be registered to each of the trained atlas, and segmented using the equation 2. Consequently,  $N_a$  different MLMapping segmentations can be produced for a given subject image. These label images can then be combined using any method from the multi-atlas segmentation fusion literature.

Note that these combination methods, such as majority or weighted vote, were often inspired from the multi-classifiers combination methods in the machine learning community. With the formalism proposed in this work, the atlases are now real classifiers that can be combined using these methods.

In the experiment section, we will see that the multi-atlas version of our method indeed improve the segmentation quality over the single atlas version. We will see that as opposed to standard label fusion method, non-local fusion is not beneficial for the combination of the output segmentations. So a simple weighted vote, which can be very efficiently done, is sufficient for the label fusion.

### III. CALCULATION

The method presented in section II is general and the different components of the method can be chosen independently depending on the application, the imaging modality, the object within the image, etc. The three main components of our framework are: the registration method, the classifier and the feature function. When the multi-atlas MLMapping is used, the multi-atlas combination method should also be defined.

#### A. Image Registration

As in the training step, the transformation output of the registration needs to be inverted, the registration method should be able to produce such transform. Many published registration methods are now able to do so. The registrations in this work have been performed using our previously published registration method [48] and inverted as in [5]. The transformation is modeled using B-spline and constraints during the registration force the Jacobian to be positive for every voxel in the image.

Note also that, as in [4], the inclusion of the neighbors of the matching point in the segmentation process allows us to be much less accurate in the registration. We also tested our method using only rigid transformations. These transforms are more robust and faster to compute but much less accurate.

#### B. Local Features

Numerous local image descriptors have been proposed in the literature for image matching, image recognition, texture characterization, etc. Our method is not bound

to the choice of the feature vector but in our current implementation, the most straightforward local descriptor is used: the local feature is a vector whose components are the intensities of the voxels in a box shaped neighborhood of a given voxel. This is similar to the patch approach uses for example in [4] or [23]. We followed the recommendation of [4] and took 3x3x3 as the size of the box defining the feature. Note that, when a linear classifier is used with local patch as feature vector, as we do, it is not necessary to add any linear filter output such as local mean or smoothed derivatives to the feature vector. The classification step will be in charge to find the most discriminant linear combination of the voxel in the patch. To be robust to intensity variation or inhomogeneity artifacts, the images have been corrected for bias field inhomogeneities, they have been histogram equalized and rescaled in  $[0, 1]$ .

#### C. Classification Method

Two things should be kept in mind for the choice of the classifier: its evaluation for a new vector should be efficient, and storing its parameters should take a minimal size in memory. Indeed, these two requirements will enable the evaluation of equation 2 efficiently, and consequently, to have a fast segmentation once the registration is done. Ideally, we would have no overhead when computing the equation 2 compared to equation 1, this would mean that the MLMapping would have a computation time similar to standard atlas-based segmentation.

In this work, a linear support vector machine (SVM) classifier is used to define the voxels of  $C_a$ . A binary linear SVM solves the problem:

$$\min_{w,b} \frac{1}{2}w^2 + \frac{C}{N} \sum_{i=1}^N \max(0, 1 - y_i(wx_i + b)), \quad (3)$$

where  $x_i$  are the feature vectors,  $y_i \in \{-1, 1\}$  the corresponding labels,  $w$  and  $b$  define the separating hyper plane and  $C$  is a penalization parameter. A "one versus the rest" strategy is used to account for the multi-class nature of our problem: a binary classifier is trained for each class using the features from all other classes as negative points. This classifier fulfills the two criteria stated above and the overhead of MLMapping compared to atlas-based segmentation is negligible. The storage cost is reasonable: only the normal  $w$  and the offset  $b$  of the separating hyper plane for each class involved in the local classification are stored. The computation time is also low: for each class of the local classifier, the computational cost is the cost of the few dot products between the feature patch and the normal hyper plane. The main problem is in the training time. As we will see in section IV F, the training time is very high and is a limiting factor for the feature dimension or the number of atlases in the training dataset. However, the training is done offline and its cost is hidden to the user.

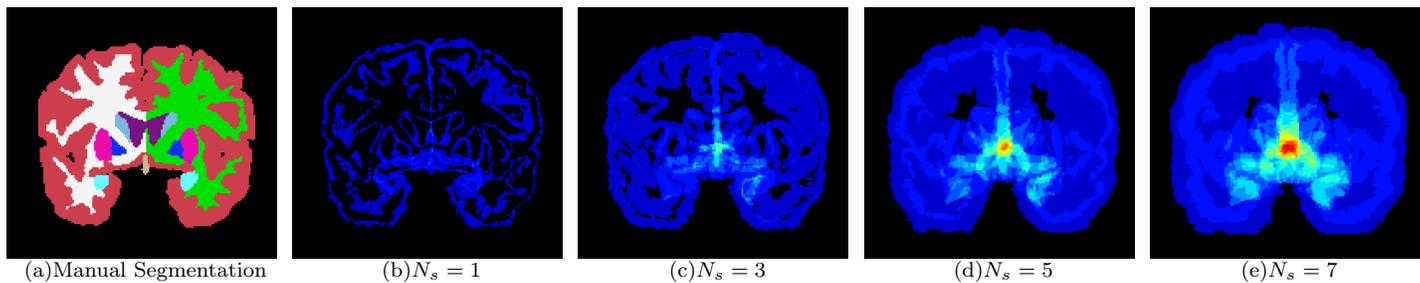


FIG. 1. Number of classes in the multi-class classifier as a function of the neighborhood size  $N_s$ . The colormap is black when the classifier is constant (one class) and varies from blue to red when the number of classes varies from 2 to 16.

Note that usually the number of classes is very low for most of the voxels in the image. In figure 1 is presented the images of number of classes as a function of the neighborhood size. In black, where only one class is involved, the cost is exactly the same as standard segmentation. In dark blue, the classifiers are binary and only one dot product is needed for the segmentation. In this example, where 5 images are used in the training dataset and when  $N_s = 5$  is set for the training, the atlas is composed at 84% of constant classifiers, 10% of binary classifiers, 4% of ternary classifiers and for less than 2% more than three class are involved.

The LIBSVM library [49] has been used for the training of the local classifiers.

#### D. MAML Multi-Atlas Combination

If the multi-atlas MLMapping is considered, a multi-atlas combination method should be used on top on the MLMapping procedure. In this work, the outputs of the classifier atlases are fused using a non-local weighted vote procedure. Formally, we define  $L_i$ , the vector image with 1 on the  $k$ th component if  $S_i(x) = k$  and 0 otherwise and the vector image  $L$  as:

$$L(x) = \frac{\sum_{i=1}^{N'_a} \sum_{y \in N_i(x)} w_i(x, y) L_i(y)}{\sum_{i=1}^{N'_a} \sum_{y \in N_i(x)} w_i(x, y)}, \quad (4)$$

where  $N'_a$  is the number of atlas used in the combination. The weight  $w_i$  measures the local similarities between the input image at  $x$  and the  $i$ th atlas at  $y$  as the distance between their feature vectors using a Gaussian kernel. The final label on the voxel  $x$  is the label corresponding to the highest component of  $L(x)$ . This procedure is similar to what has been proposed in [4, 29].

The parameters of the multi-atlas fusion are the  $\sigma$  parameter of the Gaussian kernel and the size of the neighborhood  $N_i$  defined as a  $N'_s \times N'_s \times N'_s$  box.

This fusion method has been evaluated either with a standard labeling (equation 1) or with the proposed MLMapping (equation 2). As we will see in the experiment section, an interesting property of the multi-atlas MLMapping is that it does not require a non-local fusion:

the optimal  $N'_s$  value is 1. This allows a non-negligible saving of the computation time for the fusion of the multiple atlases.

## IV. EXPERIMENTAL

### A. Experimental Setup

In this section, the influence of the parameters  $N_s$  and  $C$  of the single atlas MLMapping method is studied in section IV B. The multi-atlas MLMapping parameters  $N'_a$ ,  $N'_s$  and  $\sigma$  are analyzed in section IV C. Our methods are compared with the state of the art single or multi-atlas segmentation procedures in section IV D and IV E. The notation and meaning of the different method's parameters has been summarized in table I.

When only one atlas is used for the registration, the two mappings are denoted as STD for the standard atlas-based registration (equation 1) and ML for the MLMapping (equation 2). In both cases, each image of the training dataset is successively chosen as the atlas to segment images of the test set. For the STD method, the manual segmentation is used as the label image to be mapped, no fusion is done on the label images beforehand.

The multi-atlas MLMapping method of section II D is denoted as MAML. It will be compared to a non-local weighted vote denoted as MASTD. For both methods, the fusion is performed using equation 4 with a  $3 \times 3 \times 3$  patch as feature vector, a  $N'_s \times N'_s \times N'_s$  box shaped neighborhood, and a weighting of each voxel using a Gaussian function with a parameter  $\sigma$ . The difference between MAML and MASTD is that in equation 4,  $L_i$  are generated from the manual segmentation of the  $i$ th atlas for MASTD whereas for MAML, it is generated from the output of the classifiers. MASTD is similar to the method of Coupé or Rousseau [4, 29]. The results will also be compared with the widely used FreeSurfer software [34].

The experiments have been performed using the IBSR dataset provided by the Center for Morphometric Analysis at Massachusetts General Hospital[50]. It is composed of the images and manual segmentations into 32 structures of the brain of 18 healthy subjects. The

|          |  |
|----------|--|
| $N_s$    | Neighborhood size for the training of the classifier images        |
| $N'_s$   | Neighborhood size for the multi-atlas fusion                       |
| $N_a$    | Number of atlases in the dataset for the classifier image training |
| $N'_a$   | Number of atlases in the multi-atlas fusion (equation 4)           |
| $C$      | SVM Penalization coefficient (equation 3)                          |
| $\sigma$ | Gaussian parameter for the multi-atlas fusion (equation 4)         |

TABLE I. Notations

structures used in this work are given in the table IX of the appendix. As for symmetric structures the results obtained are similar for the left and right parts, only the left structures results are reported. The images are  $256 \times 256 \times 128$  MR  $T_1$  images with a resolution of  $0.93 \times 0.93 \times 1.5$ ,  $1 \times 1 \times 1.5$  or  $0.83 \times 0.83 \times 1.5$  mm<sup>3</sup>. Note that FreeSurfer has its own 40 images training dataset. However, the FreeSurfer dataset have been created using the same manual segmentation protocol as the IBRSR dataset.

Cross validation has been done by randomly splitting the database in three: 5 subjects were used as the training dataset, 4 subjects were used to tune the parameters of the different methods while the remaining data were used as a test set for comparison. The tuning set is used to find the optimal value for: the  $C$  penalization of the SVM (equation 3), the neighborhood size  $N_s$  in the MLMapping training and the  $\sigma$  and the neighborhood size  $N'_s$  of the multi-atlas fusion (equation 4). The validation protocol has been constrained by the computational time of the training as we will see in section IV F: while the segmentation time of our method is low, the training time is not and is problematic for the experiments with a large training dataset.

A quantitative comparison of the different methods has been done using the Dice metric and the average Hausdorff distance between the ground truth (G) and the automatic segmentation (A). For a given region of interest (ROI), the Dice score is given by:

$$D(G, A) = 100 \times \frac{2|G \cap A|}{|G| + |A|},$$

and the average Hausdorff distance is given by:

$$AvgDst(G, A) = \frac{\sum_{x \in G} \text{dist}(x, A)}{2|G|} + \frac{\sum_{x \in A} \text{dist}(x, G)}{2|A|}.$$

Except for FreeSurfer which has its own registration procedure, all the methods were tested with either a rigid registration or the non-rigid registration algorithm described in [48]. A B-spline node spacing of 4x4x4 has been used and the registration is run with a constraint ensuring that the Jacobian is positive for all the voxels.

An example of the results of STD, MASTD, ML and MAML is presented on figure 2.

## B. MLMapping Parameters Influence

Two parameters are crucial in the performance of MLMapping: the size  $N_s$  of the box used to include neighbors in the training and the  $C$  parameter of the SVM learning step (equation 3). They have been chosen by optimizing the mean Dice (see figure 3).

One can remark that the optimal  $C$  is dependent on the neighborhood size  $N_s$  and the size of the image dataset: it is increasing with these two parameters. This unfortunately implies that  $C$  has to be optimized for each new dataset.

## C. Multi-Atlas MLMapping Parameter Influence

To study the influence of the parameters of MAML, a classifier image has been computed for all the images of the training dataset. The training has been done with the optimal MLMapping parameters ( $N_s^{\text{opt}} = 5$  and  $C^{\text{opt}} = 10^{15}$ ) and with a suboptimal value for the MLMapping  $N_s$  parameter ( $N_s^{\text{subopt}} = 3$  and  $C^{\text{subopt}} = 10^{15}$ ).

The segmentation has then been performed with several values of the multi-atlas combination parameters: the neighborhood size  $N'_s$  and the  $\sigma$  parameter of the Gaussian weighting function. The mean Dice of MAML for different parameters values as well as those of MASTD is presented in figure 4.

The first thing to notice is that, as opposed to MASTD, increasing of  $N'_s$  for MAML does not improve the segmentation when  $N_s^{\text{opt}}$  is used. It is even detrimental to the results. Simply, this means that the MLMapping is good at capturing the information from the neighbors. Once the optimal  $N_s$  value is chosen for the training, there is no point in looking at neighboring classifiers as they embed information from voxels too far away from the original one. With an optimal  $N_s$  for the MLMapping training, using  $N'_s > 1$  for MAML, seems to be similar to the use of a too large  $N_s$  value during the MLMapping training.

This interpretation seems to be confirmed by the figure 4(d): when  $N_s < N_s^{\text{opt}}$  for the MLMapping training, using  $N'_s > 1$  enable to catch up with the missing neighborhood information but then, when  $N'_s$  is too large, performance also decreases.

In the table II, the mean Dice of the MAML method is given as a function of the number of atlases combined,  $N'_a$ . Each atlas is trained with the whole training dataset (five labeled images),  $N_s = N_s^{\text{opt}}$  and  $N'_s = 1$ . One

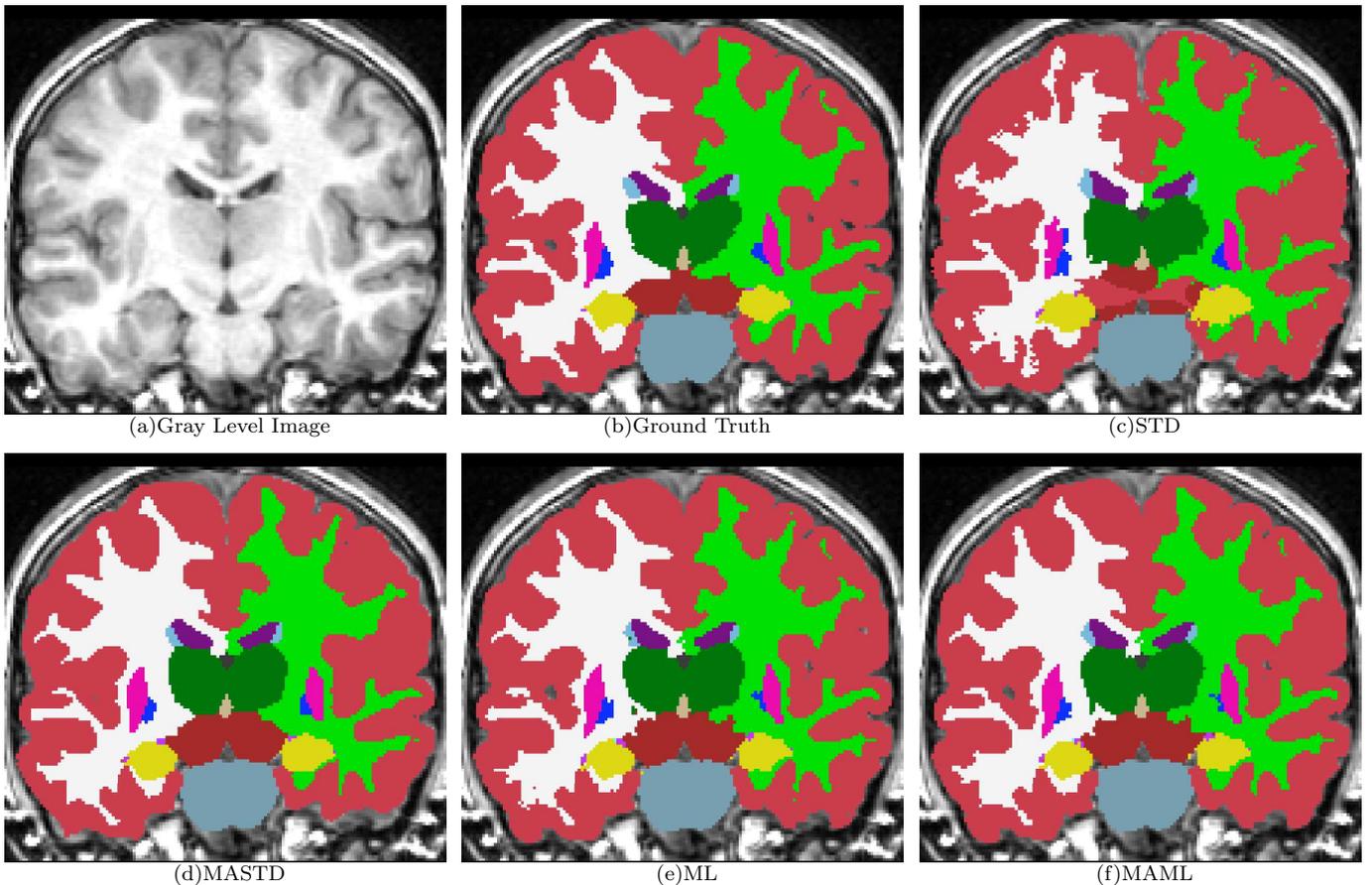


FIG. 2. An example of an image of the IBSR dataset with its manual segmentation (Ground Truth). The results of the automatic segmentation using standard atlas-based segmentation (STD), MLmapping (ML), FreeSurfer, multi-atlas standard labeling (MA-STD) and multi-atlas MLmapping (MA-ML).

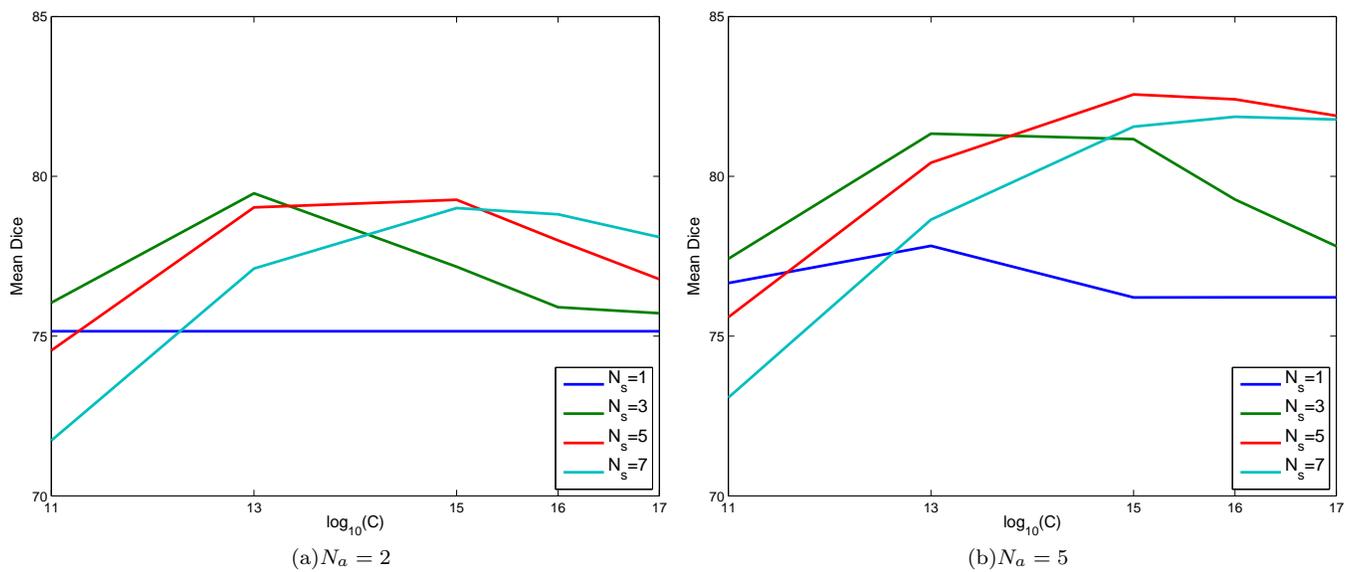


FIG. 3. Mean Dice for the comparison of ML results when the  $C$  parameter of the SVM and the learning neighborhood box size  $N_s$  are varying. Results are given when the size of the training dataset is 2 or 5.

can see that the method behaves differently when  $N'_s$  or  $N'_a$  varies: while increasing  $N'_s$  was detrimental to the

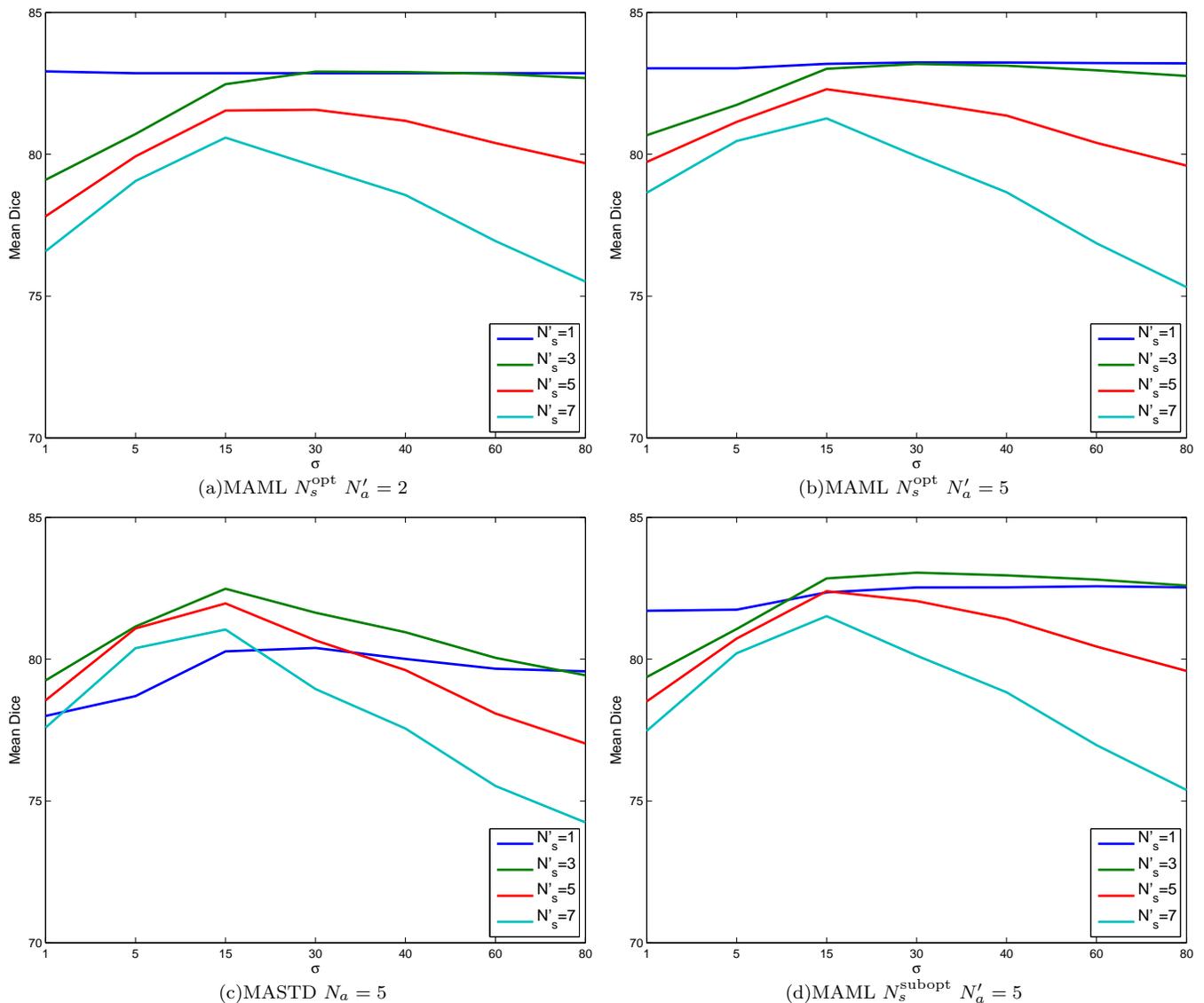


FIG. 4. Mean Dice for the comparison of ML results when the  $\sigma$  parameter of the non-local fusion and the learning neighborhood box size  $N_s$  are varying. Results are given when the size of the fusion dataset is 2 or 5.

segmentation, the addition of new voxels by the addition of atlases in the segmentation images fusion improves the results. However, one can see that the improvement is limited.

#### D. Comparison of Single Atlas Methods

In this section, methods requiring the registration to only one atlas to segment a subject are compared. We include in the comparison the standard atlas-based segmentation STD, FreeSurfer, the weighted vote with one atlas and a neighborhood of size  $N_s = 5$  (which was the optimal value when only one atlas was used) and the MLMapping with either rigid or nonlinear registration. MLMapping has been used with the optimal parameters

found in the previous section ( $N_s^{\text{opt}} = 5$ ,  $C^{\text{opt}} = 10^{15}$  for ML-Nonlinear and  $N_s^{\text{opt}} = 7$ ,  $C^{\text{opt}} = 10^{17}$  for MLRigid). Structure by structure Dice values and mean Dice are reported in the table III, the average Hausdorff distances are reported in table IV. In these tables, the two best methods are in bold, the stars indicate the best one.

As one can see, both the rigid and nonrigid version of MLMapping clearly outperforms the other methods, with a clear advantage when a nonlinear registration is used. ML-Nonlinear outperforms STD by 8 points on the mean Dice. It also has a 5 points difference with weighted vote and 4 points difference with FreeSurfer. ML-Nonlinear is consistently the best method for all structures except cerebellum white matter for which FreeSurfer is better. Indeed, even if the registration to only a single atlas is used for the segmentation using ML, this atlas carries the

| $N'_a$    | 1     | 2     | 3     | 4     | 5     |
|-----------|-------|-------|-------|-------|-------|
| Mean Dice | 79.83 | 80.12 | 80.51 | 80.45 | 80.63 |

TABLE II. Mean Dice for MAML as a function of the number of classifier atlases used for the segmentation.

| ROI        | STD       | STD+ $N_s = 5$      | FreeSurfer          | ML-Rigid         | ML-Nonlinear        |
|------------|-----------|---------------------|---------------------|------------------|---------------------|
| LCaWM      | 82.3± 5.6 | 88.3± 6.1           | <b>88.9± 5.2</b>    | 88.7±10.4        | * <b>90.1± 9.3*</b> |
| LCaCo      | 83.2± 5.9 | 88.7± 5.3           | 83.1±22.3           | <b>89.5±12.4</b> | * <b>92.4± 5.3*</b> |
| LLatVe     | 84.9±19.2 | * <b>91.6±12.8*</b> | 78.2±29.0           | 87.1±21.5        | <b>91.2±17.5</b>    |
| LInflLatVe | 28.2±12.9 | 34.5±17.6           | 40.9±24.2           | <b>44.4±15.6</b> | * <b>49.7±14.6*</b> |
| LCuWM      | 71.8±25.8 | 68.9±43.9           | * <b>83.9± 8.8*</b> | 75.2±35.3        | <b>77.1±35.1</b>    |
| LCuCort    | 83.4±18.2 | 84.6±24.7           | 87.4±10.2           | <b>89.0±13.4</b> | * <b>91.2±11.4*</b> |
| LThPr      | 83.7±12.9 | 86.8± 8.8           | <b>86.9± 4.2</b>    | 85.6±11.3        | * <b>87.5±11.3*</b> |
| LCau       | 77.8±10.1 | <b>83.5± 5.6</b>    | 83.4± 7.7           | 83.3±12.8        | * <b>85.8± 6.0*</b> |
| LPut       | 81.5± 8.6 | 82.2± 7.0           | 82.5± 5.8           | <b>84.9±13.6</b> | * <b>88.0± 7.3*</b> |
| LPal       | 67.9±15.2 | 73.8±17.9           | 75.2±26.2           | <b>78.8± 7.3</b> | * <b>80.1± 8.3*</b> |
| 3Ve        | 67.9±15.9 | 72.3±14.0           | <b>79.3± 9.0</b>    | 78.3±11.5        | * <b>79.5±14.2*</b> |
| 4Ve        | 70.7±11.8 | 73.6±13.4           | <b>76.6±15.8</b>    | 75.8±10.0        | * <b>78.0±11.4*</b> |
| BStem      | 88.9± 5.8 | <b>90.3± 7.1</b>    | 86.4± 4.1           | 90.2± 8.5        | * <b>92.2± 6.3*</b> |
| LHippo     | 69.7±11.5 | 72.0±14.5           | <b>73.7± 8.7</b>    | 72.3±23.0        | * <b>77.8±11.1*</b> |
| LAmyg      | 59.7±16.6 | 61.7±17.8           | <b>65.5± 9.3</b>    | 64.9±16.3        | * <b>67.2±13.7*</b> |
| CSF        | 48.5±28.2 | 56.2±31.9           | 57.5±21.9           | <b>58.8±29.9</b> | * <b>59.7±32.7*</b> |
| LAcc       | 58.8±20.8 | 58.1±19.4           | 59.1± 7.8           | <b>60.9±16.8</b> | * <b>67.0±16.4*</b> |
| LVeDC      | 76.8±12.6 | <b>80.7±12.8</b>    | 73.6± 8.3           | 79.2±15.1        | * <b>81.5±14.9*</b> |
| Mean Dice  | 71.4±14.3 | 74.8±15.5           | 75.6±12.6           | <b>77.0±15.8</b> | * <b>79.7±13.7*</b> |

TABLE III. Single Atlas Segmentation Comparison: average Dice values for each structures of interest for single atlas segmentation methods: standard atlas-based segmentation (STD), standard atlas-based segmentation with neighbors contributions (STD+ $N_s = 5$ ), FreeSurfer, MLMapping with rigid registration (ML-Rigid) and MLMapping with a deformable registration (ML-Nonlinear). Rigid registration was used for ML-Rigid, FreeSurfer deformable registration was used for FreeSurfer, for all the other methods, the deformable registration described in [48] was used. The two best methods are in bold, the star indicates the best one.

| ROI        | STD       | STD+ $N_s = 5$      | FreeSurfer          | ML-Rigid            | ML-Nonlinear        |
|------------|-----------|---------------------|---------------------|---------------------|---------------------|
| LCaWM      | 0.24±0.04 | 0.15±0.03           | <b>0.14±0.03</b>    | 0.16±0.07           | * <b>0.12±0.05*</b> |
| LCaCo      | 0.22±0.05 | <b>0.14±0.03</b>    | 0.19±0.09           | 0.17±0.11           | * <b>0.08±0.02*</b> |
| LLatVe     | 0.29±0.34 | * <b>0.10±0.04*</b> | 0.31±0.26           | 0.23±0.21           | <b>0.15±0.25</b>    |
| LInflLatVe | 1.59±0.51 | 2.48±0.84           | 3.30±1.93           | <b>1.05±0.39</b>    | * <b>0.95±0.34*</b> |
| LCuWM      | 0.51±0.32 | 0.97±0.96           | * <b>0.25±0.08*</b> | 0.58±0.51           | <b>0.48±0.46</b>    |
| LCuCort    | 0.29±0.19 | 0.28±0.23           | <b>0.15±0.04</b>    | 0.21±0.13           | * <b>0.13±0.07*</b> |
| LThPr      | 0.23±0.12 | 0.18±0.06           | * <b>0.16±0.02*</b> | 0.22±0.08           | <b>0.17±0.08</b>    |
| LCau       | 0.29±0.08 | <b>0.19±0.02</b>    | 0.21±0.04           | 0.23±0.10           | * <b>0.16±0.03*</b> |
| LPut       | 0.25±0.07 | 0.24±0.06           | 0.23±0.05           | <b>0.22±0.10</b>    | * <b>0.14±0.05*</b> |
| LPal       | 0.46±0.15 | 0.36±0.16           | 0.35±0.25           | <b>0.29±0.09</b>    | * <b>0.25±0.07*</b> |
| 3Ve        | 0.51±0.40 | 0.32±0.10           | <b>0.29±0.08</b>    | 0.29±0.07           | * <b>0.28±0.11*</b> |
| 4Ve        | 0.48±0.18 | 0.39±0.12           | 0.43±0.35           | <b>0.35±0.09</b>    | * <b>0.31±0.07*</b> |
| BStem      | 0.15±0.05 | <b>0.13±0.04</b>    | 0.18±0.03           | 0.16±0.06           | * <b>0.11±0.04*</b> |
| LHippo     | 0.47±0.18 | 0.47±0.20           | <b>0.36±0.07</b>    | 0.48±0.27           | * <b>0.32±0.14*</b> |
| LAmyg      | 0.63±0.25 | 0.68±0.29           | <b>0.48±0.08</b>    | 0.58±0.23           | * <b>0.48±0.17*</b> |
| CSF        | 1.80±1.70 | <b>1.26±1.01</b>    | 1.26±0.94           | * <b>1.22±0.92*</b> | 1.33±1.27           |
| LAcc       | 0.61±0.34 | 0.64±0.34           | <b>0.57±0.13</b>    | 0.76±0.36           | * <b>0.54±0.30*</b> |
| LVeDC      | 0.32±0.11 | <b>0.26±0.09</b>    | 0.37±0.07           | 0.30±0.13           | * <b>0.26±0.12*</b> |
| Mean       | 0.52±0.28 | 0.51±0.26           | 0.51±0.25           | <b>0.42±0.22</b>    | * <b>0.35±0.20</b>  |

TABLE IV. Single Atlas Segmentation Comparison: average Hausdorff distance values for each structures of interest for single atlas segmentation methods: standard atlas-based segmentation (STD), standard atlas-based segmentation with neighbors contributions (STD+ $N_s = 5$ ), FreeSurfer, MLMapping with rigid registration (ML-Rigid) and MLMapping with a deformable registration (ML-Nonlinear). Rigid registration was used for ML-Rigid, FreeSurfer deformable registration was used for FreeSurfer, for all the other methods, the deformable registration described in [48] was used. The two best methods are in bold, the star indicates the best one.

information of the whole training dataset.

The worst results are obtained by the standard atlas-

| ROI       | MV        | MAML-Rigid         | MASTD              | ML-Nonlinear     | MAML-Nonlinear     |
|-----------|-----------|--------------------|--------------------|------------------|--------------------|
| LCaWM     | 86.6± 5.6 | 88.5±10.0          | 89.3± 9.9          | <b>90.1± 9.3</b> | <b>*90.6±10.2*</b> |
| LCaCo     | 86.9± 6.0 | 89.3±12.0          | 89.9± 7.3          | <b>92.4± 5.3</b> | <b>*92.9± 5.6*</b> |
| LLatVe    | 90.1±10.7 | 88.0±22.2          | <b>92.1±12.0</b>   | 91.2±17.5        | <b>*92.6±12.2*</b> |
| LInFLatVe | 33.5± 7.0 | 37.7±24.2          | 46.4±22.0          | <b>49.7±14.6</b> | <b>*52.4±15.6*</b> |
| LCuWM     | 75.5±29.7 | 74.4±37.0          | 76.1±36.4          | <b>77.1±35.1</b> | <b>*77.5±36.9*</b> |
| LCuCort   | 86.4±18.6 | 89.4±14.0          | 88.1±19.9          | <b>91.2±11.4</b> | <b>*91.8±11.0*</b> |
| LThPr     | 86.9±10.2 | 86.2±12.0          | <b>*87.9±10.5*</b> | 87.5±11.3        | <b>87.9±10.8</b>   |
| LCau      | 82.0± 7.5 | 84.2±14.2          | <b>86.0± 6.0</b>   | 85.8± 6.0        | <b>*86.2± 6.6*</b> |
| LPut      | 85.8± 5.9 | 85.4±14.8          | 87.8± 6.0          | <b>88.0± 7.3</b> | <b>*88.3± 7.3*</b> |
| LPal      | 75.2±14.8 | 78.9± 9.3          | 77.8±12.4          | <b>80.1± 8.3</b> | <b>*80.9± 7.4*</b> |
| 3Ve       | 74.7±14.6 | <b>*80.0±11.8*</b> | 78.9±17.0          | 79.5±14.2        | <b>79.8±14.9</b>   |
| 4Ve       | 77.3± 9.6 | 75.4±11.6          | 77.5±14.7          | <b>78.0±11.4</b> | <b>*78.9±12.7*</b> |
| BStem     | 91.1± 5.3 | 90.7± 8.2          | 91.6± 6.5          | <b>92.2± 6.3</b> | <b>*92.4± 6.4*</b> |
| LHippo    | 75.9±10.1 | 74.4±25.5          | <b>79.0±11.8</b>   | 77.8±11.1        | <b>*79.4±10.7*</b> |
| LAmyg     | 66.2±18.0 | 67.2±18.3          | <b>67.9±16.4</b>   | 67.2±13.7        | <b>*68.8±14.2*</b> |
| CSF       | 54.7±30.2 | 58.9±30.3          | 57.5±33.3          | <b>59.7±32.7</b> | <b>*60.5±33.6*</b> |
| LAcc      | 66.5±19.2 | 62.9±18.7          | <b>*70.6±13.5*</b> | 67.0±16.4        | <b>68.6±16.2</b>   |
| LVeDC     | 81.4±12.8 | 79.8±16.0          | <b>*82.1±15.2*</b> | 81.5±14.9        | <b>81.8±15.7</b>   |
| Mean Dice | 76.493    | 77.295             | 79.250             | <b>79.780</b>    | <b>*80.630*</b>    |

TABLE V. Multi Atlas Segmentation Comparison: average Dice values for each structures of interest for single atlas MLMapping (ML-Nonlinear) and multi-atlas segmentation methods: Majority Voting (MV), Multi Atlas MLMapping with a rigid registration (MAML-Rigid), Non local Multi Atlas patch weighted vote (MASTD), MLMapping with deformable registration (ML) and Multi Atlas MLMapping with a deformable registration (MAML-Nonlinear). Rigid registration was used for MAML-Rigid, the deformable registration described in [48] was used for all the other methods. The two best methods are in bold, the star indicates the best one.

| ROI       | MV               | MAML-Rigid         | MASTD              | ML-Nonlinear       | MAML-Nonlinear     |
|-----------|------------------|--------------------|--------------------|--------------------|--------------------|
| LCaWM     | 0.17±0.03        | 0.17±0.08          | 0.13±0.05          | <b>0.12±0.05</b>   | <b>*0.11±0.05*</b> |
| LCaCo     | 0.16±0.04        | 0.17±0.12          | 0.12±0.04          | <b>0.08±0.02</b>   | <b>*0.08±0.02*</b> |
| LLatVe    | 0.12±0.04        | 0.21±0.20          | <b>0.09±0.04</b>   | 0.15±0.25          | <b>*0.08±0.04*</b> |
| LInFLatVe | 1.25±0.21        | 1.73±1.10          | 1.11±0.46          | <b>*0.95±0.34*</b> | <b>0.97±0.54</b>   |
| LCuWM     | <b>0.48±0.39</b> | 0.75±0.76          | 0.55±0.53          | <b>*0.48±0.46*</b> | 0.49±0.50          |
| LCuCort   | 0.23±0.16        | 0.19±0.13          | 0.20±0.16          | <b>0.13±0.07</b>   | <b>*0.11±0.06*</b> |
| LThPr     | 0.17±0.06        | 0.21±0.10          | <b>*0.16±0.07*</b> | 0.17±0.08          | <b>0.16±0.07</b>   |
| LCau      | 0.21±0.03        | 0.22±0.14          | <b>0.16±0.02</b>   | 0.16±0.03          | <b>*0.15±0.03*</b> |
| LPut      | 0.17±0.04        | 0.21±0.12          | <b>0.14±0.04</b>   | 0.14±0.05          | <b>*0.14±0.05*</b> |
| LPal      | 0.31±0.11        | 0.29±0.11          | 0.28±0.08          | <b>0.25±0.07</b>   | <b>*0.23±0.05*</b> |
| 3Ve       | 0.33±0.20        | <b>*0.26±0.08*</b> | 0.27±0.15          | 0.28±0.11          | <b>0.27±0.09</b>   |
| 4Ve       | 0.33±0.08        | 0.36±0.09          | 0.33±0.09          | <b>0.31±0.07</b>   | <b>*0.29±0.08*</b> |
| BStem     | 0.11±0.03        | 0.14±0.06          | <b>0.10±0.03</b>   | 0.11±0.04          | <b>*0.10±0.03*</b> |
| LHippo    | 0.32±0.08        | 0.45±0.33          | <b>0.28±0.09</b>   | 0.32±0.14          | <b>*0.26±0.07*</b> |
| LAmyg     | 0.48±0.25        | 0.54±0.28          | <b>0.46±0.22</b>   | 0.48±0.17          | <b>*0.44±0.18*</b> |
| CSF       | 1.32±1.09        | <b>1.25±0.98</b>   | 1.26±1.08          | 1.33±1.27          | <b>*1.22±1.05*</b> |
| LAcc      | <b>0.43±0.23</b> | 0.69±0.38          | <b>*0.39±0.14*</b> | 0.54±0.30          | 0.47±0.23          |
| LVeDC     | 0.25±0.09        | 0.30±0.15          | <b>0.25±0.11</b>   | 0.26±0.12          | <b>*0.25±0.12*</b> |
| Mean      | 0.380            | 0.454              | 0.348              | <b>0.347</b>       | <b>*0.325*</b>     |

TABLE VI. Multi Atlas Segmentation Comparison: average Hausdorff distance values for each structures of interest for single atlas MLMapping (ML-Nonlinear) and multi-atlas segmentation methods: Majority Voting (MV), Multi Atlas MLMapping with a rigid registration (MAML-Rigid), Non local Multi Atlas patch weighted vote (MASTD), MLMapping with deformable registration (ML) and Multi Atlas MLMapping with a deformable registration (MAML-Nonlinear). Rigid registration was used for MAML-Rigid, the deformable registration described in [48] was used for all the other methods. The two best methods are in bold, the star indicates the best one.

based segmentation STD. It is indeed very sensitive to the registration performance. This method is clearly improved by simply adding the contribution of the voxels in the neighborhood of the matching point (STD+ $N_s = 5$ ).

Note that even if FreeSurfer also carries information from several atlases (40 images) it is outperformed by MLMapping (with a training set of 5 images) which seems much better at capturing the information required for the seg-

mentation.

Note however that as FreeSurfer has its own deformable registration procedure, it is difficult to tell if its results can be improved by changing the registration or its classification routine.

One can reach similar conclusions when looking at average Hausdorff distances in table IV as we did for the Dice values. It can however be noticed that, although MLMapping with nonlinear registration still outperforms the other methods, it is now not the best method for three structures, and the rigid MLMapping loses its second rank for two structures. The use of the average Hausdorff distance seems to be (slightly) beneficial to FreeSurfer and  $STD + N_s = 5$ .

Between STD and ML-nonlinear, the average Hausdorff decrease ranges from 5% to 63% depending on the structure with an average of 35% reduction per structure. Between  $STD + N_s = 5$  and ML-nonlinear, the average Hausdorff is increased of 50% for the Lateral Ventricle. Apart from this structure, the results range from a 5% increase to a 61% decrease with an average of 22% improvement per structure.

### E. Comparison of Multi-Atlas Methods

In table V the Dice values for each structure and the mean Dice of the four multi-atlas methods MV (Majority Voting), MASTD, MAML with either rigid or non-rigid registration are reported. The results of ML with nonlinear registration, which is a single atlas method, are also reported for comparison. The average Hausdorff distances for each structure are reported in table VI. Optimal parameters found in the section IV B and IV C have been used:  $N'_s = 1$ ,  $\sigma = 40$  for MLMapping,  $N'_s = 3$ ,  $\sigma = 15$  for MASTD.

Out of the 18 structures of interest, the nonlinear registration version of MAML has the best Dice value for 14 structures. For three structures (the accumbens area, the ventral diencephalon and the thalamus proper), MASTD gives the better results. This confirms that MAML-Nonlinear gives a segmentation of high quality. One can also notice that despite the fact that only one registration is required, ML is second when looking at the mean Dice as it ranked second for many structures. The segmentation quality of ML is similar to MASTD while only one registration is required and no time is needed for the combination of several segmentation images. The worse results are obtained with MV and MAML-Rigid. One can notice that although only a rigid registration is used for MAML-Rigid it is competitive with MV which uses a deformable registration.

As one can see in table VI, when the metric is the average Hausdorff distance, the ranking of the methods is similar. One can notice that although MAML-Nonlinear is still ranked first on most structures (and second otherwise) its superiority is less significant: it is first on eleven structures while it was first on 14 structures with the

Dice metric.

Between MASTD and MAML-nonlinear, the average Hausdorff distance is increased by 22% for the Accumbens area. Apart from this structure, the average Hausdorff ranges from a 3% increase to a 34% decrease with an average of 8% improvement per structure.

### F. CPU time consideration

The main limitation of MLMapping is its training time. This time depends on three parameters: the classifier parameters ( $C$ ), the size of the training set ( $N_a$ ) and the size of neighborhood ( $N_s$ ). CPU time of the training and segmentation time have been measured on a HP computer with Intel Xeon CPU E5620 2.40GHz processor and reported in the table VII and VIII. As one can see in the table VII, for a training dataset of five images, this time is reasonable when a nonlinear registration and optimal parameters are used: five hours when  $N_a = 5$ ,  $N_s = 5$  and  $C = 10^{15}$ . However, the computation time is a real problem when a rigid registration and the optimal  $N_s$  are used: the time needed to train one atlas with the optimal parameters ( $N_s = 7$ ,  $C = 10^{17}$ ) is 411 hours for a dataset of five atlases. Such a computation time is acceptable for a one time training of one atlas, it is however a problem for the numerical experimentation and cross validation involving repeated training on several atlases. For the user of the atlas, this problem does not exist: once the learning has been done, the atlas can be used at the cost of the registration used.

Although a large time is required for the training, the advantage of MLMapping resides in the time required for the segmentation itself. The table VIII presents a comparison between the cost of ML, MAML and MASTD. Note that the reported time does not account for the registration, the values reflect the fusion time only. So the CPU cost of the registration, linear in  $N'_a$ , should be added to get the total time. Depending on the image size, the image content, the transformation and the algorithm this time usually range from few minutes to more than an hour and sometimes several hours per registration. While the computational time depends on the  $N'_s$  and  $N'_a$  for MASTD it depends only on  $N'_a$  for MAML and is constant for ML mapping. This is a clear advantage of MLMapping especially when a linear registration is used. In this case, with only 5 atlases, the time post registration is reduced from almost an hour for MASTD to few seconds for ML or few dozens of seconds for MAML.

Note that if the transformation between a chosen template and all the atlases are precomputed and stored, the MASTD method can also be implemented with only one registration to the template: the transformation from the subject to any atlas is obtained by composition as in [28]. Even in this case, ML is advantageous over MASTD as the non-local label fusion time is null for ML and few dozens of minutes for MASTD.

| $N_s$ | $N_a = 2$     |               |               | $N_a = 5$     |               |               |
|-------|---------------|---------------|---------------|---------------|---------------|---------------|
|       | $C = 10^{13}$ | $C = 10^{15}$ | $C = 10^{17}$ | $C = 10^{13}$ | $C = 10^{15}$ | $C = 10^{17}$ |
| 1     | : :09         | : :06         | : :07         | : :26         | : :20         | : :21         |
| 3     | : 3: 52       | : 3: 17       | : 7:27        | :19: 59       | :22: 04       | 8:03:13       |
| 5     | :55: 06       | :47: 56       | 20:48:19      | 5:06: 01      | 5:05: 35      | 106:24:27     |
| 7     | 5:27: 48      | 5:32: 24      | 107:32:21     | 39:06: 19     | 35:25: 14     | 411:38:12     |

TABLE VII. CPU time (hours:minutes:seconds) for the training of one classifier image as a function of the number of atlas ( $N_a$ ) and the size of the neighborhood ( $N_s$ ) and the  $C$  parameter of the support vector machine.

| $N'_a$ | MASTD      |            |            |            | ML    | MAML |
|--------|------------|------------|------------|------------|-------|------|
|        | $N'_s = 1$ | $N'_s = 3$ | $N'_s = 5$ | $N'_s = 7$ |       |      |
| 1      | : 03       | :59        | 3:23       | 11:19      | 00:02 | :02  |
| 2      | : 08       | 1:42       | 6:49       | 19:12      |       | :07  |
| 3      | : 10       | 3:06       | 13:06      | 25:18      |       | :11  |
| 4      | : 16       | 4:31       | 19:58      | 33:12      |       | :15  |
| 5      | : 17       | 4:30       | 19:06      | 50:08      |       | :18  |

TABLE VIII. CPU time (minutes:seconds) for the segmentation using a weighted vote (MASTD) with  $N'_a$  atlases or ML mapping. The time reported account only for the fusion, not the (potentially multiple) registration needed before the fusion.

## V. CONCLUSION

In this work, a new method to improve the atlas-based segmentation method has been presented. The idea is to change the mapping of the segmentation labels to the subject to segment once the registration has been done. An atlas is now defined by a gray level image and an image of local classifiers. For a given voxel in the subject image, a local feature is computed and given as input into the classifier of the corresponding voxel in the atlas that will assign the output label. This new formalism generalizes the standard mapping for which all classifiers are constant and always return the corresponding manual segmentation label of the atlas. As opposed to other methods of the literature, the additional computation overhead due to the incorporation of new information is offline, during the training of the classifier image.

The limitation of our method is the training time which can be very long, especially for the optimization of the parameter and the setup of the numerical experimentation. This training time constraint is however completely transparent for the atlas user as the training is done once and for all. This aspect surely deserves more investigations and will be the object of a future work. Another limitation of the proposed method is the cost required to store the classifier images. Indeed, for each trained atlas, one has to store several binary classifiers per voxel (except when there is only one class for this voxel though). This constrains us in the choice of the classifier: it should be represented using a reasonable number of parameters. This also constrains us in the choice of the feature vector: for linear SVM, the number of parameters of the model is the size of the feature vector. This problem can potentially be addressed using a feature selection step and will be investigated in a future work.

The ML mapping presented in this work allows segmenting an image at the cost of standard atlas-based segmentation but with a quality similar to multi-atlas segmentation. As in [4, 29], neighborhood information is also included in our framework but, in our case, its cost is also completely offline. Further enhancement is still possible when several atlases are registered to the subject image.

It is assumed in this work, that a standard registration preprocessing has been applied to the images. Standard preprocessing can include artifact correction and/or intensity normalization.

In this paper, the feature vector used is a local patch. One of the interest of a linear classifier (such as linear SVM) with a local patch as feature vector is that in the training step, the most discriminant linear filter will be selected as the normal to the separating hyper plane. This may be a low pass filter when it is the raw average local intensities that will discriminate the classes. It can also be any directional derivative filter (and in this case, it is invariant of any intensity shift). It is left to the algorithm to adaptively decide the nature of the filter that is applied to find the label.

The choice of the feature vector is very important and the results can probably be improved by more investigations on the design the feature vector. One can take advantage of the huge literature in computer vision for the development on local descriptor vectors that have been successfully used for object recognition, segmentation and content based image retrieval [33, 51–53]. One can also use recent results on using local feature for shape classification or segmentation in medical images such as [54, 55]. If tissue probability maps [56] are used are features, a tissue coherent mapping can be obtained [5, 57]. This can be extended in the multi-atlas case to label probability maps that can be computed on the fly, and

also used as features.

## Appendix A: Indexes of the IBSR database ROIs

### ACKNOWLEDGMENT

We gratefully acknowledge Ms Lucy Katz for reading and correcting the manuscript.

This work was performed within the framework of the LABEX PRIMES (ANR-11-LABX-0063) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

- 
- [1] D. L. Collins, C. J. Holmes, T. M. Peters, and A. C. Evans, "Automatic 3d model-based neuroanatomical segmentation," *Human Brain Mapping*, vol. 3, no. 3, pp. 190–208, 1995.
- [2] M. Miller, G. Christensen, Y. Amit, and U. Grenander, "Mathematical textbook of deformable neuroanatomies," in *Proceedings of the National Academy of Science of the United States of America*, vol. 90, pp. 11944–8, 1993.
- [3] S. Warfield, A. Robatino, J. Dengler, F. Jolesz, and R. Kikinis, *Nonlinear Registration and Template Driven Segmentation*, ch. 4, pp. 67–84. Brain Warping, 1998.
- [4] F. Rousseau, P. Habas, and C. Studholme, "A supervised patch-based approach for human brain labeling," *Medical Imaging, IEEE Transactions on*, vol. 30, pp. 1852–1862, Oct 2011.
- [5] M. Sdika, "Combining atlas based segmentation and intensity classification with nearest neighbor transform and accuracy weighted vote," *Medical Image Analysis*, vol. 14, no. 2, pp. 219 – 226, 2010.
- [6] T. Rohlfing, R. Brandt, R. Menzel, and C. R. Maurer, "Evaluation of atlas selection strategies for atlas-based image segmentation with application to confocal microscopy images of bee brains," *NeuroImage*, vol. 21, pp. 1428–1442, 4 2004.
- [7] I. Isgum, M. Staring, A. Rutten, M. Prokop, M. Viergever, and B. van Ginneken, "Multi-atlas-based segmentation with local decision fusion - application to cardiac and aortic segmentation in CT scans," *IEEE Trans on Medical Imaging*, vol. 28, pp. 1000–1010, Jul 2009.
- [8] H. Kirişli, M. Schaap, S. Klein, S. Papadopoulou, M. Bonardi, C. Chen, A. Weustink, N. Mollet, E. Vonken, R. van der Geest, *et al.*, "Evaluation of a multi-atlas based method for segmentation of cardiac cta data: a large-scale, multicenter, and multivendor study," *Medical physics*, vol. 37, no. 12, pp. 6279–6291, 2010.
- [9] T. R. Langerak, U. A. van der Heide, A. N. Kotte, M. A. Viergever, M. van Vulpen, and J. P. Pluim, "Label fusion in atlas-based segmentation using a selective and iterative method for performance level estimation (simple)," *Medical Imaging, IEEE Transactions on*, vol. 29, no. 12, pp. 2000–2008, 2010.
- [10] M. Fotuhi, D. Do, and C. Jack, "Modifiable factors that alter the size of the hippocampus with ageing," *Nature Reviews Neurology*, vol. 8, no. 4, pp. 189–202, 2012.
- [11] O. Commowick, V. Grégoire, and G. Malandain, "Atlas-based delineation of lymph node levels in head and neck computed tomography images," *Radiotherapy and Oncology*, vol. 87, no. 2, pp. 281–289, 2008.
- [12] K. D. Fritscher, M. Peroni, P. Zaffino, M. F. Spadea, R. Schubert, and G. Sharp, "Automatic segmentation of head and neck ct images for radiotherapy treatment planning using multiple atlases, statistical appearance models, and geodesic active contours," *Medical physics*, vol. 41, no. 5, p. 051910, 2014.
- [13] J. Yang, A. Amini, R. Williamson, L. Zhang, Y. Zhang, R. Komaki, Z. Liao, J. Cox, J. Welsh, L. Court, *et al.*, "Automatic contouring of brachial plexus using a multi-atlas approach for lung cancer radiation therapy," *Practical radiation oncology*, vol. 3, no. 4, pp. e139–e147, 2013.
- [14] R. Stefanescu, O. Commowick, G. Malandain, P.-Y. Bondiau, N. Ayache, and X. Pennec, "Non-rigid atlas to subject registration with pathologies for conformal brain radiotherapy," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2004*, pp. 704–711, Springer, 2004.
- [15] C. Lu, S. Chelikani, D. Jaffray, M. F. Milosevic, L. H. Staib, J. S. Duncan, *et al.*, "Simultaneous nonrigid registration, segmentation, and tumor detection in mri guided cervical cancer radiation therapy," *Medical Imaging, IEEE Transactions on*, vol. 31, no. 6, pp. 1213–1227, 2012.
- [16] J. Carballido-Gamio, S. Bonaretti, I. Saeed, R. Harnish, R. Recker, A. J. Burghardt, J. H. Keyak, T. Harris, S. Khosla, and T. F. Lang, "Automatic multi-parametric quantification of the proximal femur with quantitative computed tomography," *Quantitative Imaging in Medicine and Surgery*, vol. 5, no. 4, p. 552, 2015.
- [17] M. Bustamante, S. Petersson, J. Eriksson, U. Alehagen, P. Dyverfeldt, C. J. Carlhall, and T. Ebbens, "Atlas-based analysis of 4D flow CMR: Automated vessel segmentation and flow quantification," *J Cardiovasc Magn Reson*, vol. 17, no. 1, p. 87, 2015.
- [18] S. K. Warfield, K. H. Zou, and I. William M. Wells, "Validation of image segmentation and expert quality with an expectation-maximization algorithm," in *Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention-Part I*, pp. 298–306, Springer-Verlag, 2002.
- [19] T. Rohlfing, D. B. Russakoff, and C. R. Maurer, Jr., "Performance-based classifier combination in atlas-based image segmentation using expectation-maximization parameter estimation," *IEEE Transactions on Medical Imaging*, vol. 23, no. 8, pp. 983–994, 2004.
- [20] T. Rohlfing and C. R. Maurer, Jr., "Shape-based averaging," *IEEE Transactions on Image Processing*, vol. 16, pp. 153–161, Jan. 2007.
- [21] R. A. Heckemann, J. V. Hajnal, P. Aljabar, D. Rueck-

|   |           |                              |    |       |                      |    |        |                     |
|---|-----------|------------------------------|----|-------|----------------------|----|--------|---------------------|
| 2 | LCaWM     | Left Cerebral White Matter   | 10 | LThPr | Left Thalamus Proper | 16 | BStem  | Brain Stem          |
| 3 | LCaCo     | Left Cerebral Cortex         | 11 | LCau  | Left Caudate         | 17 | LHippo | Left Hippocampus    |
| 4 | LLatVe    | Left Lateral Ventricle       | 12 | LPut  | Left Putamen         | 18 | LAMyg  | Left Amygdala       |
| 5 | LInfLatVe | Left Inf Lat Vent            | 13 | LPal  | Left Pallidum        | 24 | CSF    | CSF                 |
| 7 | LCuWM     | Left Cerebellum White Matter | 14 | 3Ve   | 3rd Ventricle        | 26 | LAcc   | Left Accumbens area |
| 8 | LCuCort   | Left Cerebellum Cortex       | 15 | 4Ve   | 4th Ventricle        | 28 | LVeDC  | Left VentralDC      |

TABLE IX. Index / Abbreviation / Full Name correspondence table of the IBSR dataset structures used in the experiments.

- ert, and A. Hammers, “Automatic anatomical brain MRI segmentation combining label propagation and decision fusion,” *NeuroImage*, vol. 33, no. 1, pp. 115 – 126, 2006.
- [22] T. Rohlfing and C. R. Maurer, Jr., “Multi-classifier framework for atlas-based image segmentation,” *Pattern Recognition Letters*, vol. 26, no. 13, pp. 2070–2079, 2005.
- [23] X. Artaechevarria, A. Munoz-Barrutia, and C. Ortizde Solorzano, “Combination strategies in multi-atlas image segmentation: application to brain MR data,” *IEEE Trans Med Imaging*, vol. 28, pp. 1266–1277, Aug 2009.
- [24] M. R. Sabuncu, B. T. Yeo, K. Van Leemput, B. Fischl, and P. Golland, “A generative model for image segmentation based on label fusion,” *Medical Imaging, IEEE Transactions on*, vol. 29, no. 10, pp. 1714–1729, 2010.
- [25] J. Eugenio Iglesias, M. Rory Sabuncu, and K. Van Leemput, “A unified framework for cross-modality multi-atlas segmentation of brain mri,” *Medical image analysis*, vol. 17, no. 8, pp. 1181–1191, 2013.
- [26] H. Wang, J. W. Suh, S. R. Das, J. B. Pluta, C. Craige, and P. A. Yushkevich, “Multi-atlas segmentation with joint label fusion,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 3, pp. 611–623, 2013.
- [27] C. Ledig, R. Wolz, P. Aljabar, J. Lotjonen, R. A. Heckemann, A. Hammers, and D. Rueckert, “Multi-class brain segmentation using atlas propagation and em-based refinement,” in *Biomedical Imaging (ISBI), 2012 9th IEEE International Symposium on*, pp. 896–899, IEEE, 2012.
- [28] M. Depa, G. Holmvang, E. J. Schmidt, P. Golland, and M. R. Sabuncu, “Towards efficient label fusion by pre-alignment of training data,” in *Medical image computing and computer-assisted intervention: MICCAI... International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 14, p. 38, NIH Public Access, 2011.
- [29] P. Coupé, J. V. Manjón, V. Fonov, J. Pruessner, M. Robles, and D. L. Collins, “Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation,” *NeuroImage*, vol. 54, no. 2, pp. 940–954, 2011.
- [30] A. Buades, B. Coll, and J.-M. Morel, “A review of image denoising algorithms, with a new one,” *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
- [31] S. Liao, Y. Gao, and D. Shen, “Sparse patch based prostate segmentation in ct images,” in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2012*, pp. 385–392, Springer, 2012.
- [32] C. Wachinger, M. Brennan, G. C. Sharp, and P. Golland, “On the importance of location and features for the patch-based segmentation of parotid glands,” in *MICCAI Workshop on Image Guided Adaptive Radiation Therapy*, pp. 63–70, 2014.
- [33] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.
- [34] B. Fischl, D. H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. van der Kouwe, R. Killiany, D. Kennedy, S. Klaveness, A. Montillo, N. Makris, B. Rosen, and A. M. Dale, “Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain,” *Neuron*, vol. 33, pp. 341–355, Jan 2002.
- [35] S. Powell, V. A. Magnotta, H. Johnson, V. K. Jammalamadaka, R. Pierson, and N. C. Andreasen, “Registration and machine learning-based automated segmentation of subcortical and cerebellar brain structures,” *Neuroimage*, vol. 39, no. 1, pp. 238–247, 2008.
- [36] E. Y. Kim, V. A. Magnotta, D. Liu, and H. J. Johnson, “Stable atlas-based mapped prior (stamp) machine-learning segmentation for multicenter large-scale mri data,” *Magnetic resonance imaging*, vol. 32, no. 7, pp. 832–844, 2014.
- [37] H. Wang, S. R. Das, J. W. Suh, M. Altinay, J. Pluta, C. Craige, B. Avants, and P. A. Yushkevich, “A learning-based wrapper method to correct systematic errors in automatic image segmentation: consistently improved performance in hippocampus, cortex and brain segmentation,” *NeuroImage*, vol. 55, no. 3, pp. 968–985, 2011.
- [38] F. van der Lijn, M. de Bruijne, S. Klein, T. den Heijer, Y. Hoogendam, A. van der Lugt, M. Breteler, and W. Niessen, “Automated brain structure segmentation based on atlas registration and appearance models,” *Medical Imaging, IEEE Transactions on*, vol. 31, pp. 276–286, Feb 2012.
- [39] D. Zikic, B. Glocker, and A. Criminisi, “Encoding atlases by randomized classification forests for efficient multi-atlas label propagation,” *Medical Image Analysis*, vol. 18, no. 8, pp. 1262 – 1273, 2014. Special Issue on the 2013 Conference on Medical Image Computing and Computer Assisted Intervention.
- [40] Y. Hao, T. Wang, X. Zhang, Y. Duan, C. Yu, T. Jiang, and Y. Fan, “Local label learning (lll) for subcortical structure segmentation: application to hippocampus segmentation,” *Human brain mapping*, vol. 35, no. 6, pp. 2674–2697, 2014.
- [41] W. Bai, W. Shi, C. Ledig, and D. Rueckert, “Multi-atlas segmentation with augmented features for cardiac mr images,” *Medical image analysis*, vol. 19, no. 1, pp. 98–109, 2015.
- [42] H. Wang, Y. Cao, and T. Syeda-Mahmood, “Multi-atlas segmentation with learning-based label fusion,” in *Machine Learning in Medical Imaging* (G. Wu, D. Zhang, and L. Zhou, eds.), vol. 8679 of *Lecture Notes in Computer Science*, pp. 256–263, Springer International Publishing, 2014.
- [43] D. G. Lowe, “Distinctive image features from scale-

- invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [44] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [45] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.
- [46] S. Durrleman, M. Prastawa, J. R. Korenberg, S. Joshi, A. Trounev, and G. Gerig, “Topology preserving atlas construction from shape data without correspondence using sparse parameters,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pp. 223–230, Springer, 2012.
- [47] P. Kochunov, J. L. Lancaster, P. Thompson, R. Woods, J. Mazziotta, J. Hardies, and P. Fox, “Regional spatial normalization: toward an optimal target,” *Journal of computer assisted tomography*, vol. 25, no. 5, pp. 805–816, 2001.
- [48] M. Sdika, “A fast nonrigid image registration with constraints on the Jacobian using large scale constrained optimization,” *Medical Imaging, IEEE Transactions on*, vol. 27, no. 2, pp. 271–281, 2008.
- [49] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.
- [50] <http://www.cma.mgh.harvard.edu/ibsr/>.
- [51] T. Ojala and M. Pietikäinen, “Unsupervised texture segmentation using feature distributions,” *Pattern Recognition*, vol. 32, no. 3, pp. 477–486, 1999.
- [52] E. Tola, V. Lepetit, and P. Fua, “A fast local descriptor for dense matching,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.
- [53] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” *Computer Vision–ECCV 2010*, pp. 778–792, 2010.
- [54] A. A. Qazi, V. Pekar, J. Kim, J. Xie, S. L. Breen, and D. A. Jaffray, “Auto-segmentation of normal and target structures in head and neck ct images: a feature-driven model-based approach,” *Medical physics*, vol. 38, no. 11, pp. 6160–6170, 2011.
- [55] C. McIntosh, I. Svistoun, and T. G. Purdie, “Groupwise conditional random forests for automatic shape classification and contour quality assessment in radiotherapy planning,” *Medical Imaging, IEEE Transactions on*, vol. 32, no. 6, pp. 1043–1057, 2013.
- [56] Y. Zhang, M. Brady, and S. Smith, “Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm,” *IEEE Transactions on Medical Imaging*, vol. 20, no. 1, pp. 45–57, 2001.
- [57] L. D. Collins, A. P. Zijdenbos, W. F. C. Baare, and A. C. Evans, “Animal+insect: Improved cortical structure segmentation,” in *IPMI ’99: Proceedings of the 16th International Conference on Information Processing in Medical Imaging*, pp. 210–223, Springer-Verlag, 1999.