Published in final edited form as:
J Signal Process Syst. 2009 January 1; 54(1-3): 183–203. doi:10.1007/s11265-008-0243-1.

A Scalable Framework For Segmenting Magnetic Resonance Images

Prodip Hore,
Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA

Lawrence O. Hall,
Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA

Dmitry B. Goldgof,
Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA

Yuhua Gu,
Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA

Andrew A. Maudsley, and
School of Medicine, University of Miami, Coral Gables, FL, USA

Ammar Darkazanli
School of Medicine, University of Miami, Coral Gables, FL, USA

Prodip Hore: phore@csee.usf.edu; Lawrence O. Hall: hall@csee.usf.edu; Dmitry B. Goldgof: goldgof@csee.usf.edu; Yuhua Gu: yuhua@cse.usf.edu; Andrew A. Maudsley: AMaudsley@med.miami.edu; Ammar Darkazanli: ADarkazanli@med.miami.edu

Abstract

A fast, accurate and fully automatic method of segmenting magnetic resonance images of the human brain is introduced. The approach scales well allowing fast segmentations of fine resolution images. The approach is based on modifications of the soft clustering algorithm, fuzzy c-means, that enable it to scale to large data sets. Two types of modifications to create incremental versions of fuzzy c-means are discussed. They are much faster when compared to fuzzy c-means for medium to extremely large data sets because they work on successive subsets of the data. They are comparable in quality to application of fuzzy c-means to all of the data. The clustering algorithms coupled with inhomogeneity correction and smoothing are used to create a framework for automatically segmenting magnetic resonance images of the human brain. The framework is applied to a set of normal human brain volumes acquired from different magnetic resonance scanners using different head coils, acquisition parameters and field strengths. Results are compared to those from two widely used magnetic resonance image segmentation programs, Statistical Parametric Mapping and the FMRIB Software Library (FSL). The results are comparable to FSL while providing significant speed-up and better scalability to larger volumes of data.

Correspondence to: Lawrence O. Hall, hall@csee.usf.edu.

Keywords

Magnetic resonance imaging; Clustering; Segmentation; Automatic; Scalable; FCM

1 Introduction

The accurate, automated segmentation of magnetic resonance (MR) images provides the possibility of improved medical care. It is expected that the resolution of the MR images will increase as image acquisition technology continues to improve. In this paper, we present a framework for segmenting MR images of the human brain. Inhomogeneity (bias) correction is done on the front-end and smoothing is done on the back end of processing. In between, a robust soft clustering approach is used because such methods have been shown to provide stable and meaningful clusters [14]. In particular, fuzzy c-means (FCM) has been used as a part of several image segmentation systems of both normal and abnormal brains and other MR images [14,56–64].

For clinical use, segmentation results need to be obtained in a timely manner. FCM and other soft clustering algorithms such as EM [35] are very time consuming. They may take on the order of hours for a large volume of data and assume that all the data fits into local memory. In the clinic, you may not have large memory machines available and we may need to have scalable algorithms. Towards this end, a number of methods to scale up FCM have been developed [20,24,26–28], but all require the full data set reside in memory. We introduce two approaches to create a fast, scalable version of FCM. The result is a segmentation package (FSSP) that is faster and significantly more scalable (in terms of features and voxels) than two widely used packages FMRIB Software Library (FSL) [1] and Statistical Parametric Mapping (SPM) [2].

In [31] we introduced a variant of FCM which makes one pass through the data by clustering subsets of the data and creating a final partition by using weighted centroids which represent the subsets. We showed that this single pass fuzzy c-means algorithm (SPFCM) provides clustering quality which is very close to that of FCM given different random initializations even when we load as little as 1% of the data at a time. It allows for clustering of data sets which are too large for memory, but also allows for fast clustering of data that fits in memory.

In this paper, we introduce and compare the performance of an online fuzzy clustering algorithm called online FCM (OFCM) [32]. Unlike other single pass and scalable algorithms [21,25,30,31] it can produce good segmentation quality without randomly accessing data. The data is viewed as a stream and can come in any order. The approach introduced here might be viewed as a modification of a streaming algorithm to summarize the entire data set through a set of evolving cluster centroids.

The MR image segmentation algorithms we have developed are a part of the MIDAS Project [53], whose goal is to simplify implementation of Magnetic Resonance Spectroscopic Imaging (MRSI) for routine diagnostic imaging studies and to map normal metabolite distributions in human brains. As part of the package, we use an existing bias correction algorithm to deal with inhomogeneity and a spatial smoothing process. The fast clustering algorithms enable the potential clinical use of the approach. MIDAS, and hence our approach, is designed to work on any MR scanner, with any head coil, with any sequence, with minimal tuning and in an automated way. The segmentation step is used with the spectroscopic imaging and software for automated MRSI processing, brain region mapping, statistical analysis, and clinical presentation. On completion, this software suite will be used

for diagnostic neuroimaging applications including cancer, epilepsy, and neurodegenerative disease. More about the MIDAS project is available in [50–52]. We will show that our segmentation approach is comparable to FSL, which has been shown to be the current best package in [65] when compared to 6 other approaches. FSSP is significantly faster than FSL or SPM.

2 Related Work

Magnetic resonance images (MRI) have been widely used in medical applications. MR imaging provides non-invasive detailed images of living tissues. This paper is focused on MR brain images from normal human brains. MRI may be used for defining an anatomical framework for functional visualization, cortical surface mapping, volume measurement, tissue classification, and morphological adaptation assessment. Brain image segmentation plays a very important role. In particular, segmenting the brain image into cerebro-spinal fluid (CSF), grey matter (GM), and white matter (WM) is extremely important for quantitative analysis and can serve as a baseline for MR spectroscopy.

In the literature, there exists a variety of segmentation schemes. A review of methods for brain image segmentation was done by Pham [3]. The paper discussed the advantages and disadvantages of the segmentation approaches used for medical imaging applications. However, many of the approaches had the problem of required human interaction for accurate and reliable results. Automation is necessary to process large amounts of brain data. On the other hand, fully automated segmentation systems often cannot provide accurate tissue classification due to their sensitivity to issues, such as the partial volume effect and inter-tissue intensity contrast reduction. For these reasons, many classical intensity-based classification models, such as the fuzzy-c means model and the mixture of Gaussians model may give unstable results with tissue region misclassification caused by the presence of artifacts. This area of research has received a large amount of attention. Many methods have been proposed [4–13].

Conventional classification models often have another drawback. Most of them are based on two-dimensional (2-D) images, which means processing is done slice by slice and eventually 2D results are combined into 3D. This has benefits but has other issues. To guarantee the continuity of segmentation and the integrity of tissues along the third dimension, one must not just ignore the neighboring slices when doing individual slice processing, however 3-D methods are often computationally intensive. If the slices are processed along the axial plane (X and Y), the third dimension will be the Z axis. It is also necessary to map tissues along the third dimension to make a 3-D segmentation model. Mapping tissue in the Z direction may not be a trivial task. In our previous work [14], a slice based automatic brain image segmentation expert system showed some of the drawbacks of using slice processing.

Many 3D algorithms have also been developed [15,16], but volumetric processing of large data sets tends to require a long time. FSL [1] and SPM [2] are currently publicly available popular packages for doing brain image analysis; they both contain good segmentation tools. Zhang [17] provided a method to improve segmentation smoothness and immunity to noise; a finite mixture (FM) model was used for statistical segmentation of brain MRI, but later it was found that FM's had a critical limitation, it only worked on well-defined images with low noise. So, for real data with artifacts such as inhomogeneity and bias field distortion, a problem occurs. Thus, a finite model by itself cannot produce reliable results in this case. In order to overcome the difficulties, Zhang [17] took into account spatial information. They incorporated both the Hidden Markov Random Fields (HMRF) model and the EM algorithm into a HMRF-EM framework to solve the inhomogeneity problem at the same time. They also incorporated the bias field correction algorithm of Guillemaud and Brady [19] into the

model. This fully automated 3-D model proved to be robust and reliable for normal brain segmentation.

The segmentation routine in SPM was described in [18] and works as follows; the MRI is fit into an MRI template by means of a 12-parameter affine transformation. For each voxel of the template, there exists a three-element vector specifying the a priori probabilities for GM, WM, and CSF. Then the probability map is used to segment the fit MRI into its four partitions (GM, WM, CSF, and "everything else.") using an iterative soft clustering scheme. Recently, in [65] a study of seven different automatic segmentation tools was reported. Because ground truth is expensive and difficult to obtain, they proposed a method of evaluating the performance of seven segmentation algorithms using a common agreement principle framework. They also compared limited regions of the human brain to manually drawn ground truth and got similar ratings of the algorithms to those obtained by using the common agreement principle, which uses no ground truth. It was reported that FSL using one feature (T1) was best overall. Although SPM (using one feature) was not always the best, its rating was good and was close to the performance of FSL. Again, SPM like the FSL package does a good job in segmenting brain images. Both are widely used in segmenting normal MRI brain images; however, everyone using these two packages will observe that both of them require quite significant computational time when processing a whole brain volume. This is, arguably, the main disadvantage of using these two packages.

Many speed up techniques based on sampling [21,25] have been proposed but for large data sets even a representative sample may be big enough so that it does not fit in memory. Recently, many single pass algorithms have been proposed [29–31,33–36] which create a final partition by scanning the data on the disk once. Single pass algorithms are an attractive choice for scaling because they work under the notion of limited memory allocated and are known to be fast and produce good quality final partitions. We believe future scalable approaches should be based on either single pass or incremental or online clustering algorithms. This is because besides scalability, these algorithms or their variants will provide the framework for real time processing.

brFCM [22] speeds up clustering by reducing the resolution of the data. However, it requires that all data be memory resident. In [42,43], a streaming algorithm was proposed using a k-Median clustering algorithm called LOCALSEARCH using a single pass view of the data. Data arrived in chunks and then each chunk was clustered using a LOCALSEARCH algorithm and the memory was freed by summarizing the clustering result by weighted centroids. This is somewhat similar to our OFCM algorithm though more complex as data is treated hierarchically and the number of clusters may require time consuming search. A main difference is in the fact that in fuzzy clustering an example may not completely belong to a particular cluster. Our method of summarizing clustering results involves the fuzzy membership matrix, which does not exist for the crisp cases. We believe, condensing clustering solutions using fuzzy centroids and a fuzzy membership matrix is important; otherwise, the summarized clustering solutions will simply reduce to a possibly less stable form of a crisp clustering solution.

Although, the algorithm in [42] was classified as a streaming algorithm, in [37] it was pointed out that a streaming algorithm may not be viewed as single pass clustering problem. This is because a single pass algorithm over an entire stream will be dominated by outdated history. They proposed a framework for analysis of clusters over different time frames. They stored the summary statistics of the streaming data periodically using micro-clusters which were the online component of their algorithm, and later analyzed these cluster summary statistics over a user provided time horizon. Thus, the algorithm is neither purely online nor fully automatic. Recently, various other streaming algorithms [37–41] for an evolving

distribution have been proposed. In [54,55], we also introduced a streaming variant of the fuzzy-c-means algorithm for clustering evolving data streams. As stated earlier, our OFCM was designed to produce partition quality as good as clustering the full data set. Unlike recently proposed streaming algorithms, we are not concerned about clustering quality for different time horizons, but with summarizing clustering quality over the whole time horizon so that at the end the final clustering solution is as good as that obtainable from clustering the full data set.

3 Single Pass Fuzzy C Means Algorithm

Suppose we intend to cluster a large or very large data set present on a disk. We assume for large or very large data sets that the data set size exceeds the memory size. As in [30], data is randomly reordered on the disk. We can only load a certain percentage of the data based on the available memory. If we load 1% of the data into memory at a time then we have to do it 100 times to scan through the entire data set. We call each such data access a partial data access (PDA). The number of PDAs will depend on how much data we load each time. In our approach, after the first PDA, data is clustered into c partitions using fuzzy c means. Then the data in memory is condensed into c weighted points and clustered with new points loaded in the next PDA. We call them “weighted” points because they have weights, which are calculated by summing the membership of examples in a cluster. This is the key difference from the crisp clustering case [30], where a fuzzy membership matrix is not present. In each PDA new singleton points are loaded into memory and clustered along with the past *c* weighted points obtained from the previous clustering. We call this partial data clustering (PDC). After clustering these new singleton points along with the past c weighted points, they are condensed again into c new higher weighted points and clustered with examples loaded in the next PDA. This continues until all the data has been scanned once. The objective function of fuzzy c means was modified in a fashion similar to that in [22] to accommodate the effect of weights.

As an example, consider a large or very large data set of n examples. If n_1 examples are fetched in the first PDA and clustered into c partitions then all n_1 examples in memory are condensed into c weighted points, whose weights sum up to n_1. Condensation of n_1 examples into c weighted points frees the memory. Next n_2 examples are loaded into memory in the next PDA. These new n_2 examples are then clustered along with the c weighted points. So, after the second PDA there will be $n_2 + c$ examples in memory for clustering, out of which c are weighted points and n_2 examples have weight one (singletons). We will call the modified fuzzy c means algorithm which takes into account the weights of the c weighted points, weighted fuzzy c means (WFCM). After clustering these $n_2 + c$ examples in memory using WFCM they are condensed again into c new weighted points. This time the weight of the c points sum up to $n_1 + n_2$ and thus they have more weight than before. This is because there were already c weighted points, of total weight n_1, present when n_2 new singleton examples were loaded in the second PDA. Similarly, after completion of clustering in the third PDA, the weight of the new condensed c points will sum up to $n_1 + n_2 + n_3$. This means after the m^{th} PDA there will be n_m singleton points loaded in the memory along with c weighted points from a previous PDC, whose weights sum up to $n_1 + n_2 + n_3 + \ldots + n_{m-1}$. So, if the last PDA loads n_l examples, it essentially clusters the whole data set, where $n - n_l$ examples remain as c condensed weighted points and n_l as singleton points. Thus, our simple SPFCM algorithm will produce a partition in a single pass through the whole data set discarding a chunk of data after a partition is created. To speed up clustering, we initialize each PDC with the final centroids obtained from the previous PDC. This knowledge propagation allows for faster convergence.

In other work [31,32], it has been shown for SPFCM that using 10% of the data per pass the maximum difference from FCM applied to all of the data in terms of J_m was 0.3% for eight data sets. The speedups ranged from 3.3 times to 13.6 times. These results are predicated on data being reasonably well mixed by class. In the case that data comes in an order, perhaps by class, the difference can be as large as 20% from FCM. However, in [32] it was also the case that there was almost no change and a 2% change for two data sets. Clearly, the order can have an effect.

3.1 Weighted Fuzzy C Means

We modified FCM (similar to [22]) to work with weighted examples. The objective function (J_m) minimized by the WFCM is defined as follows:

$$J_m(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} w_k D_{ik}(x_k, v_i) \tag{1}$$

The cluster membership matrix and centroids for the WFCM are calculated as:

$$u_{ij} = \left[\sum_{l=1}^{c} \left(\frac{\|x_j - v_i\|}{\|x_j - v_l\|} \right)^{\frac{2}{m-1}} \right]^{-1} \tag{2}$$

$$v_i = \frac{\sum_{j=1}^{n} w_j (u_{ij})^m x_j}{\sum_{j=1}^{n} w_j (u_{ij})^m} \tag{3}$$

Where,

u_{ik} : is the membership value of the k^{th} example, x_k, in the i^{th} cluster.

v_i : is the i^{th} cluster centroid.

n : is the number of examples.

c : is the number of clusters.

w_k : is the weight of the k^{th} example.

$D_{ik}(x_k, v_i) = \|x_k - v_i\|^2$: is the norm. We have used the Euclidean distance.

It should be noted that modification of the objective function does not change the convergence property of FCM because an integer weighted example is equivalent to many examples with identical feature values (i.e. identical examples) [23]. In case of FCM, $w_k = 1$, $\forall k$. We will discuss weighted example calculation for the single pass algorithm below.

Consider n_d examples which are loaded in memory in the d^{th} PDA.

3.1.1 Case 1: d=1—If d is equal to one i.e. the first PDA, there will be no previous weighted points. In this case WFCM will be the same as FCM. In this case all n_d points have weight 1 because no weighted points from a previous PDC exist. Now, memory is freed by

condensing the clustering result into c weighted points, which are represented by the c cluster centroids v_i, where $1 \le i \le c$ and their weights are computed as follows:

$$w_i' = \sum_{j=1}^{n_d} (u_{ij}) w_j, \quad 1 \le i \le c,\ w_j = 1, \forall 1 \le j \le n_d. \tag{4}$$

The weight of the c points, after condensing the clustering results, in memory is as follows:

$$w_i = w_i', \qquad 1 \le i \le c. \tag{5}$$

It should be noted that when n_d new singleton points (weight one) are loaded in all subsequent PDA (d>1), their indices associated with $\vec{w}$ will begin at $c + 1$ and end at $n_d + c$ i.e.

$$w_j = 1, \forall c < j \le n_d + c. \tag{6}$$

3.1.2 Case 2: d>1—In this case, clustering will be applied on singleton points freshly loaded in the d^{th} PDA along with c weighted points obtained after condensation from the $(d-1)^{th}$ PDC. So, there will be $n_d + c$ points in memory for clustering using WFCM. The new n_d singleton points have weight one. After clustering, the data in memory (both singletons and weighted points) is condensed into c new weighted points. The new weighted points are represented by the c cluster centroids v_i, where $1 \le i \le c$ and their weights are computed as follows:

$$w_i' = \sum_{j=1}^{n_d + c} (u_{ij}) w_j, \qquad 1 \le i \le c. \tag{7}$$

Then memory is freed up and the weight of the condensed clustering result in memory is updated as follows:

$$w_i = w_i', \qquad 1 \le i \le c. \tag{8}$$

4 Online Fuzzy C Means

Consider a data set that has a maximum of c classes. Due to the constraints of limited memory and computation time, an online algorithm will be able to load only a small amount of data at a time depending upon the speed of data arrival and hardware capability. As in [42], we assume data is both arriving and processed in chunks (equal to the buffer size), that is, n_1 data points arrive at time t_1, n_2 at t_2, and so on. Then in the worst case a given chunk of data might come from one class only and in the best case data might come from all c classes. So, if we set the number of clusters to be always c (highest resolution) what effect will it have on different mixtures?

Case A: If less than c classes come in a chunk, then we are overclustering. Overclustering may not cause any information loss as it will likely split homogeneous clusters. Information loss occurs when we undercluster, which must group non-alike examples together.

Case B: If exactly all c classes come in a chunk, then we are neither overclustering nor underclustering.

So, setting the number of clusters always equal to c, the maximum number of classes in the data set, may not cause any information loss. Hence, we set number of clusters to be c in each chunk. After clustering data arriving at each time instant by FCM (a chunk), memory is freed by condensing the clustering solution in memory into c weighted points. The c weighted points are represented by the c cluster centroids obtained after clustering. The summarization of clustering results involves the U matrix, which indicates the fuzziness of the example's membership in each cluster. This is one key difference in our summarization from crisp algorithms.

For example if there are n_k examples in memory at time instant t_k then after clustering, memory is freed by summarizing the clustering result by c weighted centroids, whose weights are calculated using the membership matrix as follows:

$$w_i = \sum_{j=1}^{n_k} u_{ij}, 1 \le i \le c \tag{9}$$

The weighted centroids are saved on the disk. At the end, the weighted centroids of all chunks form an ensemble of weighted clustering solutions. The ensemble is then merged into the final c clusters. The merging operation is done by clustering all the weighted centroids in the ensemble using their weights. WFCM was used for this purpose. The only difference was that there were no singleton points. To speed up clustering, we initialized the cluster centroids of each chunk with the final centroids obtained from clustering the previous chunk. This knowledge propagation allows for faster convergence, provided the distribution does not change rapidly, which may often be the case.

The size of the ensemble is not likely to be large because it consists of only weighted centroids. If in any case it becomes large, similar to [42] the weighted centroids from the ensemble can be incrementally loaded and reclustered into c weighted centroids, which can be retained in memory. This will decrease the ensemble size, which can be merged at the last step into c partitions in memory. It should be noted that while clustering each chunk, we do not use any history in OFCM. This allows each chunk to equally influence the final clustering solution, which is a necessary condition for obtaining clustering quality as good as clustering the entire stream at once.

OFCM resulted in a difference of between 0.17% and 3% from FCM applied to all the data in terms of J_m [32] across nine data sets. The stream was sampled in increments of 5% which is quite small. It has the advantage compared to SPFCM of being almost insensitive to data order.

5 Data Sets

All MRI data were obtained using a T1-weighted MPRAGE sequence. Of the 33 MRI volumes (or data sets) used, 32 were obtained at the University of Miami, with 18 from a 1.5-Tesla Siemens Sonata obtained with a standard head coil and interpolated resolution of 512 × 512 × 96, and 14 obtained at 3 T on a Siemens Trio using a 8-channel phased-array head coil, and with a resolution of 256 × 256 × 144. Another data set was obtained at the VA Medical Center San Francisco using a 1.5T Siemens Magnetom, and with a resolution of 192 × 256 × 160. Although all imaging studies acquired T1-weighted (T1), Proton

Density weighted (PD) and T2-weighted (T2) images, only the T1-weighted data was used in this study. Table 1 shows the image acquisition parameters of the T1 modality in detail.

One advantage of our new clustering algorithms is that they can efficiently use multiple modalities. We compared our results with FSL (released version: 3.3.7) segmentation on the T1 weighted feature only. We can also do multiple channel result comparison, but FSL is computationally expensive on multiple channels and their results on multiple channels are not known to be well studied and reported to be poor in [65]. In order to make the comparison fair and reasonable, before performing segmentation, we used Smith's [49] (http://www.fmrib.ox.ac.uk/analysis/research/bet/) brain extraction algorithm from the FSL package to remove the skull tissue first. This will ensure that both FSL and our package use the same volume of intracranial tissue for segmentation. SPM segmentations within the same intracranial mask were also compared for evaluation purposes. FSL and SPM (SPM2 version) packages are publicly available at http://www.fmrib.ox.ac.uk/fsl/index.html and http://www.fil.ion.ucl.ac.uk/spm/software/ respectively. We used SPM2 rather than SPM5 because it was significantly faster to segment a volume and the results were closer, on average, to those of FSL.

The extracranial tissues (skull, bone, fat, etc.) and air were removed using the brain extraction tool (BET2) [49]. The intracranial tissue obtained was then used for clustering. The code for the BET2 is available at http://www.fmrib.ox.ac.uk/analysis/research/bet/.

Intensity inhomogeneity for high field MRIs may cause poor segmentation of MR images. If present, it has to be corrected to get good segmentation results. We have implemented the bias correction algorithm discussed in [48] with some modifications. We choose this bias correction algorithm because it was shown to be fast and effective. In [48], an entire MR volume was pre-processed to separate data from background/noise (generally air). Then, the background/noise was filled with the average intensity of the non-background part. This was done to prevent edge blooming. In our case, this background was the extracranial part of the brain and air obtained after running the BET2 program. Another difference is that we controlled the amount of bias correction by varying the parameters (standard deviation) of the gaussian kernel used in [48], whereas in [48] nothing was mentioned about controlling the parameters of the gaussian kernel. Intensity inhomogeneity may vary by magnetic field strength and coil setting of MR acquisition hardware, so we believe different amounts of bias correction may have to be done on MRI data obtained from different field strength and coil settings. In our methods, using both the single pass and OFCM algorithm in the MIDAS project, the user can select 3 modes of bias correction; “Light bias correction”, “Medium bias correction”, and “Custom Correction”. For “Light bias correction”, the standard deviation of the gaussian kernel in [48] was set to be 0.1, 1.5, 1.5 respectively in the Z, Y, X directions. Similarly, for “Medium bias correction” the values were 2.0, 3.0, 3.0. These values were decided after empirically testing on two training volumes, one a 1.5 T volume and the other a 3 T volume. The gaussian kernel was normalized so that the maximum value (at the peak) was 1. For “Custom Correction”, the user has to supply the parameters. Higher standard deviation values mean more bias correction will be done because it will suppress more low frequencies. One may consult [48] for more detail about the bias correction algorithm.

The 1.5T data did not appear to have any significant intensity inhomogeneity and it was therefore not necessary to perform any bias correction. The FSL package has a bias correction algorithm integrated along with their clustering algorithm. Thus, the bias correction algorithm was not a separate program we could control independently. It is expected that FSL would detect low inhomogeneity and its bias correction effect would be minimal. Like FSL, SPM2 also has a bias correction algorithm integrated into the package.

Since the 3T data was acquired using phase-array detection, bias field correction was necessary and was used for all packages.

6 MRI Data Processing

After the extracranial tissues were removed by BET2 and bias correction done (if necessary), there were 3 tissues of interest left in the volume; CSF, GM, and WM. Bias correction, when done, will be specified in the experimental section. The entire volume was then partitioned using either the SPFCM or the OFCM algorithm. For SPFCM, the number of clusters was kept to be 3. For the OFCM algorithm, we know that in whatever order the data comes, the maximum number of intracranial classes will be 3 (CSF, GM, and WM) and hence set $c = 3$. For the SPFCM as stated earlier the full data set is first randomly reordered, and was fetched in chunks (buffer) of 10% of the total data set in each PDA. In case of the OFCM algorithm the data chunk size was set to be 5% of the full data set, fetched from the bottom of the brain to the top (processed as it comes). So, OFCM will process data as it comes, while SPFCM will process randomly reordered data.

The FCM algorithm used the Euclidean distance metric, which introduces a bias towards finding spherical clusters. So, we do a single iteration of post processing using the final cluster centroids obtained after running the SPFCM/OFCM algorithm. First, an initial crisp segmentation is computed using the centroids obtained from SPFCM/OFCM by simply assigning a voxel to the nearest centroid. This is equivalent to assigning a voxel to the cluster centroid with maximum membership. Once we get the crisp segmentation, the covariance matrix associated with each cluster is then computed. Each cluster is modeled as a Gaussian distribution with the final cluster centroids obtained from SPFCM/OFCM and the associated covariance matrix. Now a refined segmentation using Bayes optimal classification [47] can be obtained using the following function:

$$P\left(z_i / x\right) = \frac{p\left(x / z_i\right) P(z_i)}{p(x)} \tag{10}$$

$$p(x) = \sum_{i=1}^{c} p\left(x / z_i\right) P(z_i) \tag{11}$$

Where c is the number of classes, $P\left(z_i / x\right)$ is the posterior probability, $p\left(x / z_i\right)$ is the class conditional density function (likelihood), $P(z_i)$ the prior probability of class z_i, and $p(x)$ the evidence factor (normalized factor). As $p(x)$ is just a scale factor, the discriminant function reduces to the following [47]:

$$g_i(x) = p\left(x / z_i\right) P(z_i) \tag{12}$$

It should be noted that $g_i(x)$ is the scaled posterior probability.

As discussed above, we modeled the class conditional density function by the Gaussian density function and assumed the prior for all classes to be the same. One may take the

natural logarithm on both sides of Eq. 12 and ignoring the constant parts, the discriminant function finally reduces to:

$$d_i = \ln g_i(x) = -\frac{1}{2}(x-\mu_i)^t \sum_i^{-1}(x-\mu_i) - \frac{1}{2}\left|\sum_i\right| \tag{13}$$

Where μ_i is the mean of the Gaussian (here we modeled it by the centroids obtained from SPFCM/OFCM), $\sum_i^{-1}$ is the inverse of the covariance matrix, and $|\Sigma_i|$ the determinant of the covariance matrix. Crisp segmentation now can be obtained by assigning the voxels to the class with the highest posterior probability. We call this the segmentation refinement step.

Another single iteration post processing step, which uses spatial information from neighborhood voxels, was also added to refine the segmentation further. One of the reasons for using spatial information in clustering is to improve the visual appearance of the segmentation (improves smoothness), which might be especially helpful for data with noise and intensity inhomogeneity. A review of several algorithms using spatial information in clustering or using Hidden Markov Models can be found in [45]. In the FSL package [1], a HMRF model was used to include spatial information. We incorporate spatial information after the segmentation refinement step. For each voxel, we used a 3 dimensional neighborhood (X, Y, and Z) and refined the posterior probability using an additive model incorporating spatial information. The additive model we used is a variant of the multiplicative model used in [46] for incorporating spatial information in FCM. Our model is as follows:

$$P\left(z_i / x\right)^r = \frac{P\left(z_i / x\right)k_1 + h\left(z_i / x\right)k_2}{\sum_{j=1}^{c}\left(P\left(z_j / x\right)k_1 + h\left(z_j / x\right)k_2\right)} \tag{14}$$

Where the spatial function is $h\left(z_j / x\right) = \sum_{x' \in N(x)} P\left(z_i / x'\right)$, $N(x)$ is a 3 dimensional window, neighborhood, centered on pixel x.

$P\left(z_i / x\right)^r$ is the refined posterior probability after using neighborhood information and the final crisp segmentation was obtained by assigning the voxel to the class with the highest probability.

The parameters k_1 and k_2 control the relative importance of neighborhood information incorporated. In our experiment we chose the size of the window to be $3 \times 3 \times 3$. The parameters k_1 and k_2 were selected to be 1 and 0.05 respectively for 1.5T data. Our 3T data was smoother with a greater influence from the local neighborhood with k_2 set to 0.10. We set these parameter values after testing on 2 volumes, one for 1.5T and one for 3T data, to get reasonable visual smoothness. We treat these two volumes as training volumes and their names will be mentioned in the results section.

To reduce computational complexity, we kept both segmentation refinement and spatial information incorporation post processing steps as a single iteration operation. These post

processing operations were done by loading data incrementally to reduce the memory requirement.

7 Experimental Results and Discussion

To evaluate the quality of tissue segmentation results, one would ideally compare results with a set of real ground truth images. In our experiments, we chose FSL to be the reference frame because recently in [65] FSL, using the T1 feature, was reported to be the best compared to six other fully automatic segmentation algorithms, including SPM. Both FSL and SPM are freely available.

Segmentation quality of each tissue was evaluated on a volume basis. For each voxel, we first look at the FSL segmentation (kept as the reference) and then we look at the same voxel position in our segmented results. If both FSL and the segmentation of another algorithm classified a voxel as the same tissue, we count it as a match. We computed how many voxels matched in an entire volume and then expressed it as a percentage. We compared the result of the FSL segmentation (kept as reference) with the segmentation of the SPM2 package and our algorithm. This was done to empirically evaluate how well we match with both FSL and SPM. It will also provide insight into how the segmentations of the packages match with each other. We will both quantitatively and visually analyze segmentation quality.

In our experiments, instead of using the current version of SPM5, we used SPM2. We compared our results based on crisp segmentation (the pixel value is either 0 or 255), however, SPM provides pixel values between 0 and 255. Hence, we had to convert the original SPM images to a crisp segmentation. We have three volumes where segmentation was very poor when using SPM5, plus the running time of SPM5 was significantly more than SPM2, therefore we used SPM2. Unlike FSL and our packages, SPM2 does not separate intracranial tissues from extracranial tissues before segmentation. It uses a probability map to produce four partitions (GM, WM, CSF, and “everything else.”) using an iterative soft clustering scheme. It assigns partial volume values to each tissue type. FSL and our packages remove the extracranial tissues (skull, bone, fat, etc.) and air using the brain extraction tool, BET2. To make a fair comparison, an SPM2 segmentation after application of the intracranial mask, produced by the BET2 program of FSL on the same volume, was extracted for comparison purposes. This will ensure segmentation results within the same area of the brain were compared for all packages. The SPM segmentation was then converted to a crisp segmentation by assigning a voxel, within the intracranial mask, to the tissue with highest membership. All experimental results are reported using the T1 weighted image.

All experiments were run on a Windows XP platform with a Pentium 4 3.00 Ghz CPU, 1 GB of memory. The running time reported for the packages here includes the time from the start of a package until it produced the final segmented results. Both FSL and FSSP are written in C and compiled under gcc here (or any C compiler in general). SPM runs in MATLAB and one would expect it to be slower to execute.

In all our experiments, the 1.5 T volume names start with MN and the 3 T volume names start with Vol. As discussed earlier, we used some training volumes for setting the parameters of the bias correction program and the spatial information incorporating (post processing) program in our algorithm. We used volume MN011 as the training volume for 1.5T data and Volume Vol009 for 3T data. It should be noted that MN011 was used only for tuning parameters of the spatial information incorporating program, as bias correction was not done on the 1.5 T data sets. After testing on the 3T training volume, Vol009, we chose the “Medium Bias Correction” option for correcting inhomogeneity on all 3T volumes

before segmenting them using FSSP with the SPFCM or OFCM algorithms. The parameter setting on the training volumes were done manually to obtain visually good segmentations. For FSL and SPM, we found their default parameter settings gave visually good segmentations on the above two training volumes. So, we did not change their default settings, except the fractional intensity threshold value of the BET2 program was kept to be 0.2 for all experiments. Its value ranges from 0 to 1, and larger the value the more will be the estimate of the intracranial brain mask.

7.1 Quantitative Comparison of Segmentations

Tables 2 and 3 show the average match of the three tissues, CSF, GM, and WM, segmented by FSSP with SPFCM or OFCM, and SPM when compared to the segmentations of FSL on the 1.5 T and 3 T volumes respectively. The match is exact voxel to voxel. Table 4 shows the average running time of each of the algorithms and the speed up obtained compared to FSL. Bias correction and inhomogeneity correction times were included for all packages for 3T data. Bias correction was not done for the 1.5T data (unnecessary) for our FSSP package. This means the speed increases on the 3T data may be more informative. We note that bias correction and inhomogeneity correction are done incrementally at every iteration in FSL and therefore are difficult to extract. Performance of the algorithms on each volume is available in Tables 9, 10, 11, 12, 13 and 14 in the Appendix section. Below we analyze the results of the algorithms.

Table 2, row 2, shows the results using FSSP with the SPFCM algorithm on the 1.5 T data. The average CSF match compared to the FSL segmentation package was 84.05%. GM and WM had match values of 84.10% and 97.47% respectively. The average running time of our algorithm based on SPFCM was 1.3 minutes whereas for FSL it was 22.53 min, which means our method is about 17 times faster. The overall average match of all three tissues on all the 1.5 T data to FSL was 88.54%. Table 2, row 4, shows the result of SPM segmentation compared to FSL segmentation on the 1.5 T volumes. The average match of CSF and WM was 75.65% and 62.64% respectively. This was much lower compared to FSSP using SPFCM (FSSP–SPFCM); though a slight increase in GM was observed (3.75%). The overall average match of SPM for all three tissues on all the 1.5 T volumes was 75.38%, which is much lower compared to FSSP–SPFCM (88.54%). FSSP using SPFCM was also more than 15 times faster compared to SPM, 1.3 min versus 20.50 min. So, the segmentation results of FSL and our FSSP package using SPFCM were in more agreement than the SPM package.

Table 3, row 2, shows the result of the FSSP using SPFCM on the 3-Tesla data. Compared to the FSL segmentation, we got an average match for CSF, GM, and WM to be 85.69%, 75.37% and 98.12% respectively. We observe the GM match was not as high as in the 1.5 Tesla data. This could be due to the difference in the bias correction algorithm. As stated earlier, FSL has the bias correction algorithm integrated in their clustering algorithm and in our case it was done before clustering. SPM also has bias correction integrated. The package based on the SPFCM outperforms FSL in terms of running time again on 3 T data; 0.68 min versus 6.52 min, which is above 9 times faster. Since the image size is smaller for 3-Tesla data, the corresponding running time for FSL is much less than with 1.5 Tesla data. The overall average match for all three tissues on all 3 T data was 86.39%. So, the average overall match of all tissues on the 3T data was lower compared to the 1.5 T data. Table 3, row 4, shows the result of the SPM segmentation compared to the FSL segmentation on the 3 T volumes. Similar to the result on 1.5 T volumes, the average match of CSF and WM, 83.49% and 59.81% respectively, was much lower compared to FSSP–SPFCM; though, an increase in GM was observed (5.71%). The overall average match of SPM of all three tissues on all the 3 T volumes was 74.79%, which is much lower compared to FSSP–SPFCM (86.39%). FSSP–SPFCM was also around 10 times faster compared to the SPM

package, 0.68 min versus 7.43 min. It seems that also on 3 T data FSL and our FSSP–SPFCM package were more in agreement than the SPM package.

Table 2, row 3, shows the results from FSSP with the OFCM (FSSP–OFCM) algorithm on the 1.5 T volumes. The FSSP–OFCM had a better average match, compared with the FSL package, than the package based on SPFCM. Compared to the FSL segmentation, there was an 88.93% match for CSF, which is about 4% higher compared to the package based on SPFCM. GM and WM matched at 85.52%, 96.56% respectively. Though WM match dropped a little bit compared to FSSP–SPFCM, it did improve GM match slightly. The average running time for FSSP–OFCM was 2.09 min, which was slower than FSSP–SPFCM but significantly faster than the FSL package (above 10 times). The overall average match rate for all three tissues on the 1.5 T data was 90.33%, which is higher than the package using SPFCM. Compared to FSL, FSSP–OFCM also had a better overall match than SPM on all the three tissues, 90.33% versus 75.17%. FSSP using OFCM was also above 9 times faster than SPM.

In Table 3, row 3, we present the results of FSSP using OFCM on the 3-Tesla data. On the 3-Tesla data, when compared to FSL, again the match of CSF and GM were better than for FSSP–SPFCM; though a slight drop in the match of WM was observed. Compared to the FSL segmentation, we got an 87.30% match for CSF, which is about 1.61% higher compared to FSSP–SPFCM. GM was 77.94%, which is about 2.57% higher compared to FSSP–SPFCM, while WM match dropped by 0.98%. Again, FSSP–OFCM was slightly slower than SPFCM, but much faster than the FSL package (above 5 times). The overall match of all three tissues on all the 3 T volumes was 87.53%; a better overall match compared to SPFCM. Compared to the segmentation of FSL, FSSP–OFCM also had a better overall match percentage than SPM on all three tissues, 87.53% versus 73.97% for SPM. It was also above 5 times faster than SPM.

7.2 Visual Comparison of Segmentation

Figures 1, 2, 3, and 4 show the segmented slices of FSL, FSSP with SPFCM, FSSP with OFCM, and SPM (we also provide the unprocessed original result from SPM2). They also display the raw data from the T1 modality. For each volume, we have chosen three slices for display; a slice through the lateral ventricles (center slice), one slice from the upper part of the brain, and one slice from the lower part of the brain. The center slice of a volume generally has a butterfly shaped ventricular CSF feature, and was chosen by visual inspection. The upper slice is chosen 10 slices above the center slice, and the lower slice is chosen 10 slices below the center slice. In the segmented images, different shades represent different tissues. Figure 1 shows that the segmented tissues from FSSP using SPFCM or OFCM and FSL on the 1.5T volumes are visually more similar to each other than the segmentation produced by SPM. The match computed on all tissues, evaluated before, also validated this. The SPM segmentation does not always look reasonable. In general, WM often looks under estimated and GM looks to be over estimated. Figure 3 shows results on the 3T volumes. On 3T data, similar results were observed. Visual observation of the raw data shows that the 3T data has visibly large intensity inhomogeneity, especially at the bottom of the slices. Compared to the FSL segmentation, visual observation reveals that the relative disagreement of the FSL segmentation with our packages generally occurred mostly in the bottom half of the slices; however, the package based on OFCM matched slightly better compared to the package based on SPFCM. Quantitative evaluation earlier also supported this fact. On average we got a better match, on all three tissues, on the 1.5 T data than on the 3T data. We believe, this happened due differences in bias correction algorithm on the 3T data, which in the future could be modified.

8 Effect of Post Processing Steps

In all the results above, simple processing was performed after a volume was segmented using the FSSP with the SPFCM or OFCM algorithm. First, the segmentation refinement algorithm used a gaussian model to minimize the bias of the Euclidean distance metric towards spherical clusters. Second, spatial information from neighborhood voxels was utilized to smooth the segmented image. Tables 5, 6, 7, and 8 show the results with and without using the post processing both for the SPFCM and the OFCM algorithms on the two training volumes, MN011 (1.5T) and Vol009 (3T). Compared to FSL, the average match rate on all three tissues always improved after the post processing steps. The minimum improvement was by 1.28% on Vol009 using the OFCM algorithm, while the maximum improvement was by 3.34% on volume MN011 also using the OFCM algorithm. Because the match rate always improved compared to FSL and FSL is a widely used segmentation package and seems to provide more stable segmentations than SPM, the use of such post processing is recommended.

9 Conclusions

We have presented a fast fully automatic human brain segmentation system for MRIs. The new method, FSSP, uses scalable approaches: single pass/incremental/online fuzzy clustering algorithms to segment an entire volume of data at once. With the improvement in hardware technology more and more high resolution images will be acquired that might not be loadable into memory. In this situation, soft clustering may be expensive to run. The single pass/incremental/online algorithms partition the entire image volume by scanning the data from the disk only once, thus preventing disk accesses which will be necessary for a standard fuzzy clustering algorithm (FCM) for data larger than available memory. Compared to FCM, these algorithms are also significantly faster even if data is loadable into memory. FSSP using OFCM will scale seamlessly to data sets larger than a standard computer memory. Single pass/incremental/online algorithms could even be used in the future with hardware innovations to enable real time MR processing. We evaluated our segmented results against two well known state of the art segmentation packages, FSL and SPM, on 33 volumes of normal human brain MR images. Recently, a study of seven automatic segmentation algorithms reported FSL and SPM as good segmentation algorithms, using the T1 image, where FSL was rated the best among all. The segmentation produced by our algorithms matches fairly close to that of the segmentation produced by the FSL package, on average on all tissues the match was above 85% both on 1.5T and 3T data (Tables 9, 10, 11, 12, 13, and 14). Segmentation results of SPM were also compared to FSL. They were not as close to FSL as ours and produced some questionable segmentations. A set of figures show the subtle visual difference between FSL and our approaches, as well as sometimes distinct differences with SPM. Compared to the FSL, FSSP based on single pass/incremental/online algorithms also required significantly less computation time and matched more closely than SPM. The scalable package introduced seems promising for processing high resolution MR images.

Acknowledgments

This research was partially supported by the National Institutes of Health under grant number 1 R01 EB00822-01 and by the Department of Energy through the ASCI PPPE Data Discovery Program, Contract number: DE-AC04-76DO00789.

Appendix

Table 9

1.5 Tesla: Comparison of segmentations of SPFCM VS FSL.

Name	CSF(%)	GM(%)	WM(%)	Running time (SP FCM) (Minutes)	Running time (FSL) (Minutes)
MN011	84.68	84.28	99.28	1.54	23.39
MN012	79.34	81.432	98.65	1.71	25.41
MN014	83.36	83.54	98.69	1.37	26.41
MN015	78.34	79.60	96.75	0.97	23.36
MN016	84.44	81.57	98.70	1.42	31.45
MN017	90.23	87.78	95.75	1.3	25.32
MN018	86.04	85.0	98.62	1.61	21.45
MN019	82.02	81.96	99.46	1.25	20.47
MN020	84.21	80.45	98.6	1.91	35.48
MN021	85.09	94.14	87.02	0.94	17.41
MN022	82.04	88.16	94.79	1.65	19.49
MN023	82.36	82.13	99.00	1.28	20.47
MN024	83.36	86.73	97.35	1.16	20.44
MN025	84.02	78.63	98.08	1.5	27.41
MN026	87.04	82.55	99.7	1.26	20.38
MN027	85.96	84.78	99.19	1.19	22.01
MN028	83.36	84.73	99.37	1.21	20.55
MN029	84.86	86.23	94.89	1.08	21.43
MNLA005	86.27	84.18	98.10	0.29	5.7
AVG	84.05	84.10	97.47	1.30	22.53

Table 10

1.5 Tesla: Comparison of segmentations of SPM2 VS FSL.

Name	CSF(%)	GM(%)	WM(%)	Running time (SPM) (Minutes)	Running time (FSL) (Minutes)
MN011	74.84	94.80	66.00	19.78	23.39
MN012	72.18	94.25	62.49	18.64	25.41
MN014	63.55	89.52	61.66	22.37	26.41
MN015	79.76	91.55	59.94	22.83	23.36
MN016	69.15	88.77	75.13	23.15	31.45
MN017	81.52	85.50	56.32	18.67	25.32
MN018	77.51	93.77	65.86	20.93	21.45
MN019	74.91	90.28	65.06	18.60	20.47
MN020	77.38	89.85	58.75	21.34	35.48
MN021	33.15	73.64	66.39	23.87	17.41
MN022	67.68	87.25	60.51	19.73	19.49
MN023	72.1	93.74	63.76	22.43	20.47
MN024	85.62	81.92	62.78	19.35	20.44

Name	CSF(%)	GM(%)	WM(%)	Running time (SPM) (Minutes)	Running time (FSL) (Minutes)
MN025	86.72	89.44	53.76	20.10	27.41
MN026	81.8	87.63	69.47	22.56	20.38
MN027	94.35	79.02	55.70	21.49	22.01
MN028	79.46	90.23	68.03	23.77	20.55
MN029	75.32	89.76	63.99	24.63	21.43
MNLA005	90.28	78.14	54.55	5.30	5.70
AVG	75.65	87.85	62.64	20.50	22.53

Table 11

3 Tesla: Comparison of segmentations of SPFCM VS FSL.

Name	CSF(%)	GM(%)	WM(%)	Running time (SP FCM) (Minutes)	Running time (FSL) (Minutes)
Vol001	84.41	76.112	98.49	0.73	6.66
Vol007	88.74	75.50	97.75	0.71	7.34
Vol009	87.18	77.81	98.16	0.63	5.67
Vol011	86.20	75.69	99.20	0.67	6.62
Vol015	87.76	76.88	97.51	0.68	6.62
Vol016	85.03	76.65	97.80	0.67	6.68
Vol018	87.73	78.75	97.10	0.63	6.61
Vol044	84.55	76.82	98.05	0.86	6.37
Vol049	87.38	74.59	98.28	0.66	7.61
Vol055	83.91	74.22	96.87	0.75	6.59
Vol057	83.31	75.20	98.69	0.57	6.6
Vol060	85.08	76.27	97.95	0.69	6.31
Vol061	84.74	70.04	99.39	0.69	6.22
Vol063	83.06	70.67	98.45	0.64	5.37
AVG	85.69	75.37	98.12	0.68	6.52

Table 12

3 Tesla: Comparison of segmentations of SPM2 VS FSL.

Name	CSF(%)	GM(%)	WM(%)	Running time (SPM) (Minutes)	Running time (FSL) (Minutes)
Vol001	68.04	85.19	64.52	7.89	6.66
Vol007	88.56	83.20	62.21	8.90	7.34
Vol009	94.44	80.95	53.32	9.13	5.67
Vol011	76.78	86.93	64.19	6.78	6.62
Vol015	86.94	84.09	62.21	7.54	6.62
Vol016	84.13	86.16	57.55	6.35	6.68
Vol018	88.7	83.94	53.52	7.67	6.61
Vol044	96.34	78.41	54.69	6.79	6.37
Vol049	97.29	71.77	52.11	8.10	7.61

Name	CSF(%)	GM(%)	WM(%)	Running time (SPM) (Minutes)	Running time (FSL) (Minutes)
Vol055	76.43	82.34	59.93	6.24	6.59
Vol057	38.34	62.63	75.99	7.38	6.60
Vol060	88.89	89.74	57.75	6.89	6.31
Vol061	91.57	81.41	64.44	7.10	6.22
Vol063	92.37	78.40	54.91	7.32	5.37
AVG	83.49	81.08	59.81	7.43	6.52

Table 13

1.5 Tesla: Comparison of segmentations of OFCM VS FSL.

Name	CSF(%)	GM(%)	WM(%)	Running time (OFCM) (Minutes)	Running time (FSL) (Minutes)
MN011	86.72	84.85	99.18	1.96	23.39
MN012	85.1	83.93	97.84	2	25.41
MN014	89.29	85.79	97.74	2.25	26.41
MN015	85.55	81.25	95.55	2.06	23.36
MN016	89.8	83.7	98.03	2	31.45
MN017	93.96	87.93	94.75	2.74	25.32
MN018	89.56	86.07	98.21	2.17	21.45
MN019	88.74	83.23	99.25	1.93	20.47
MN020	87.93	82.44	97.9	2.12	35.48
MN021	90.8	93.36	84.38	2.04	17.41
MN022	88.24	89.6	92.33	2.26	19.49
MN023	89.42	84.04	98.37	2.32	20.47
MN024	87.68	87.54	96.54	1.97	20.44
MN025	89.71	82.18	96.16	2.89	27.41
MN026	90.61	85.03	99.35	2.02	20.38
MN027	92.11	85.31	98.76	2.27	22.01
MN028	87.45	85.55	99.18	1.94	20.55
MN029	89.77	89.1	93.06	1.96	21.43
MNLA005	87.2	83.91	98.13	0.76	5.7
AVG	88.93	85.52	96.56	2.09	22.53

Table 14

3 Tesla: Comparison of segmentations of OFCM VS FSL.

Name	CSF(%)	GM(%)	WM(%)	Running time (OFCM) (Minutes)	Running time (FSL) (Minutes)
Vol001	86.51	78.75	97.72	1.16	6.66
Vol007	89.51	77.73	97.1	1.26	7.34
Vol009	87.38	79.84	97.42	1.03	5.67
Vol011	86.8	77.82	98.79	1.12	6.62
Vol015	87.42	78.02	97.17	1.18	6.62

Name	CSF(%)	GM(%)	WM(%)	Running time (OFCM) (Minutes)	Running time (FSL) (Minutes)
Vol016	86.18	77.99	97.38	1.12	6.68
Vol018	88.57	80.57	96.33	1.24	6.61
Vol044	85.62	79.24	97.28	1.39	6.37
Vol049	89.97	77.04	97.49	1.78	7.61
Vol055	86.65	77.90	95.48	1.39	6.59
Vol057	85.54	78.48	97.83	1.51	6.6
Vol060	86.78	80.10	96.66	1.31	6.31
Vol061	88.65	73.93	98.61	1.23	6.22
Vol063	86.66	73.80	97.71	1.38	5.37
AVG	87.30	77.94	97.35	1.29	6.52

References

1. FMRIB Software Library. FSL package. 2007. http://www.fmrib.ox.ac.uk/fsl/
2. SPM. SPM package. 2008. http://www.fil.ion.ucl.ac.uk/spm/
3. Pham DL, Xu C, Prince JL. Current methods in medical image segmentation. Annual Review of Biomedical Engineering. 2000; 2:315–337.
4. Marroquin JL, Vemuri BC, Botello S, Calderon F, Fernandez-Bouzas A. An accurate and efficient bayesian method for automatic a segmentation of brain MRI. IEEE Transactions on Medical Imaging. 2002 August; 21(8):934–945. [PubMed: 12472266]
5. Atkins MS, Mackiewich BT. Fully automatic segmentation of the brain in MRI. IEEE Transactions on Medical Imaging. 1998; 17(1):98–107. [PubMed: 9617911]
6. Lin, P.; Yang, Y.; Zheng, CX.; Gu, JW. Proceedings of the fourth international conference on computer and information technology (CIT 04). 2004. An efficient automatic framework for segmentation of MRI brain image.
7. Legal-Ayala, HA.; Facon, J. International conference on image processing (ICIP). 2004. Automatic segmentation of brain MRI through learning by example.
8. Wu, Z.; Paulsen, KD.; Sullivan, JM, Jr. IEEE Transactions on Biomedical Engineering. Vol. 52. 2005 June. Adaptive model initialization and deformation for automatic segmentation of t1-weighted brain MRI data; p. 1128-1131.
9. Selvathi, D.; Arulmurgan, A.; Selvi, STS.; Alagappan, S. Proceedings of the sixth international conference on computational intelligence and multimedia applications (ICCIMA 05). 2005. MRI image segmentation using unsupervised clustering techniques.
10. Wang, ZM.; Song, Q.; Soh, YC. 3rd IEEE international symposium on biomedical imaging. 2006. MRI brain image segmentation by adaptive spatial deterministic annealing clustering.
11. Mayer, A.; Greenspan, H. 3rd IEEE international symposium on biomedical imaging. 2006. Segmentation of brain MRI by adaptive mean shift.
12. Song, Z.; Tustison, N.; Avants, B.; Gee, J. 3rd IEEE international symposium on biomedical imaging. 2006. Adaptive graph cuts with tissue priors for brain MRI segmentation.
13. Greenspan H, Ruf A, Goldberger J. Constrained Gaussian mixture model framework for automatic segmentation of MR brain images. IEEE Transactions on Medical Imaging. 2006 September; 25(9):1233–1245. [PubMed: 16967808]
14. Gu, Y.; Hall, L.; Goldgof, D.; Kanade, P.; Murtagh, F. IEEE international conference on systems, man and cybernetics. 2005 October. Sequence tolerant segmentation system of brain MRI; p. 2936-2943.
15. Dawant BM, Hartmann SL, Thirion JP, Maes F, Vandermeulen D, Demaerel P. Automatic 3-D segmentation of internal structures of the head in MR images using a combination of similarity and free-form transformations: Part I, methodology and validation on normal subjects. IEEE Transactions on Medical Imaging. 1999 October; 18(10):909–916. [PubMed: 10628950]

16. Baillard, C.; Hellier, P.; Barillot, C. Proceedings IEEE workshop on mathematical methods in biomedical image analysis. 2000. Segmentation of 3D brain structures using level sets and denseregistration.
17. Zhang Y, Brady M, Smith S. Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm. IEEE Transactions on Medical Imaging. 2001; 20(1):45–57. [PubMed: 11293691]
18. Kiebel SJ, Ashburner J, Poline JB, Friston KJ. MRI and PET coregistration—a cross validation of statistical parametric mapping and automated image registration. NeuroImage. 1997; 5:271–279. [PubMed: 9345556]
19. Guillemaud R, Brady JM. Estimating the bias field of MR images. IEEE Transactions on Medical Imaging. 1997; 16(3):238–251. [PubMed: 9184886]
20. Cheng TW, Goldgof DB, Hall LO. Fast fuzzy clustering. Fuzzy Sets and Systems. 1998; 93:49–56.
21. Pal NR, Bezdek JC. Complexity reduction for “large image” processing. IEEE Transactions on Systems, Man, and Cybernetics, Part B. 2002; 32(5):598–611.
22. Eschrich S, Ke J, Hall LO, Goldgof DB. Fast accurate fuzzy clustering through data reduction. IEEE Transactions on Fuzzy Systems. 2003; 11(2):262–270.
23. Bezdek JC, Hathaway RJ, Sabin MJ, Tucker WT. Convergence theory for fuzzy C-means: Counterexamples and repairs. IEEE Transactions on Systems, Man and Cybernetics. 1987; 17(5): 873–877.
24. Altman, D. IEEE FUZZY. 1999. Efficient fuzzy clustering of multi-spectral images.
25. Hathaway RJ, Bezdek JC. Extending fuzzy and probabilistic clustering to very large data sets. Journal of Computational Statistics and Data Analysis. 2006; 51:215–234.
26. Kolen JF, Hutcheson T. Reducing the time complexity of the fuzzy C-means algorithm. IEEE Transactions on Fuzzy Systems. 2002; 10:263–267.
27. Hore, P.; Hall, LO.; Goldgof, DB. IEEE international conference on systems, man and cybernetics. 2006. A cluster ensemble framework for large data sets.
28. Borgelt C, Kruse R. Speeding up fuzzy clustering with neural network techniques. Fuzzy Systems. 2003; 2:852–856.
29. Bradley, PS.; Fayyad, U.; Reina, C. Proceedings of the fourth international conference on knowledge discovery and data mining, KDD-1998. 1998. Scaling clustering algorithms to large databases; p. 9-15.
30. Farnstrom F, Lewis J, Elkan C. Scalability for clustering algorithms revisited. ACM SIGKDD Explorations. 2000; 2:51–57.
31. Hore P, Hall LO, Goldgof DB. Single pass fuzzy C means. IEEE-FUZZ. 2007
32. Hore P. Scalable frameworks and algorithms for cluster ensembles and clustering data streams. Ph.D. Dissertation, Dept. of CSE, Univ. of South Florida. 2007 May.
33. Gupta, C.; Grossman, R. Proceedings of the fourth {SIAM} international conference on data mining (SDM 04). 2004. GenIc: A single pass generalized incremental algorithm for clustering; p. 22-24.
34. Thiesson B, Meek C, Heckerman D. Accelerating EM for large databases. Machine Learning Journal. 2001; 45:279–299.
35. Bradley PS, Fayyad UM, Reina CA. Clustering very large databases using EM mixture models. ICPR. 2000; 2:76–80.
36. Karkkainen I, Franti P. Gradual model generator for single-pass clustering. ICDM. 2005:681–684.
37. Aggarwal, CC.; Han, J.; Wang, J.; Yu, PS. Proc of VLDB. 2003. A framework for clustering evolving data streams.
38. Aggarwal, CC.; Han, J.; Wang, J.; Yu, PS. Proc of VLDB. 2004. A framework for projected clustering of high dimensional data streams.
39. Yang, J. Proc of ICDE. 2003. Dynamic clustering of evolving streams with a single pass.
40. Nasraoui, O.; Cardona, C.; Rojas, C.; Gonzlez, F. Proc of ICDM. 2003. Tecno-streams: Tracking evolving clusters in noisy data streams with a scalable immune system learning model; p. 235-242.
41. Cao F, Estery M, Qian W, Zhou A. Density-based clustering over an evolving data stream with noise. SDM. 2006

42. O'Callaghan, L.; Mishra, N.; Meyerson, A.; Guha, S.; Motwani, R. Proceedings of IEEE international conference on data engineering. 2002 March. Streaming-data algorithms for high-quality clustering.
43. Guha, S.; Meyerson, A.; Mishra, N.; Motwani, R.; O'Callaghan, L. IEEE transactions on knowledge and data engineering. 2003. Clustering data streams: Theory and practice; p. 515-528.
44. Hathaway RJ, Bezdek JC. Optimization of clustering criteria by reformulation. IEEE Transactions on Fuzzy Systems. 1995; 3:241–245.
45. Tran TN, Wehrens R, Buydens LMC. Clustering multispectral images: a tutorial. Chemometrics and Intelligent Laboratory Systems. 2005; 77:3–17.
46. Chuang KS, Tzeng HL, Chen S, Wu J, Chen TJ. Fuzzy c-means clustering with spatial information for image segmentation. Computerized Medical Imaging and Graphics. 2006; 30:9–15. [PubMed: 16361080]
47. Duda, RO.; Hart, PE.; Stork, DG. Pattern classification. 2nd. New York: Wiley-Interscience; 2000.
48. Cohen MS, DuBois RD, Zeineh MM. Rapid and effective correction of RF inhomogeneity for high field magnetic resonance imaging. Human Brain Mapping. 2000; 10(4):204–211. [PubMed: 10949058]
49. Smith SM. Fast robust automated brain extraction. Human Brain Mapping. 2002 November; 17(3): 143–155. [PubMed: 12391568]
50. Rousseau, F.; Maudsley, A.; Ebel, A.; Darkazanli, A.; Weber, P.; Sivasankaran, K., et al. Evaluation of sub-voxel registration accuracy between MRI and 3D MR spectroscopy of the brain. San Diego: SPIE Medical Imaging; 2005.
51. Young K, Chen Y, Kornak J, Matson GB, Schuff N. Classification of high dimensional, multi-spectral data sets using computational mechanics. Physical Review Letters. 2005; 94:09870.
52. Maudsley AA, Darkazanli A, Alger JR, Hall LO, Schuff N, Studholme C, et al. Comprehensive processing, display, and analysis for in vivo MR spectroscopic imaging. NMR in Biomedicine. 2006; 19:492–503. [PubMed: 16763967]
53. MIDAS. MIDAS Project Website. 2008. http://midas.med.miami.edu/
54. Hore P, Hall LO, Goldgof DB. Creating streaming iterative soft clustering algorithms. NAFIPS. 2007
55. Hore P, Hall LO, Goldgof DB. A fuzzy C means variant for clustering evolving data streams. IEEE-SMC. 2007
56. Fletcher-Heath LM, Hall LO, Goldgof DB, Reed Murtagh F. Automatic segmentation of non-enhancing brain tumors in magnetic resonance images. Artificial Intelligence in Medicine. 2001; 21:43–63. [PubMed: 11154873]
57. Clark MC, Hall LO, Goldgof DB, Velthuizen R, Murtagh R, Silbiger MS. Automatic tumor segmentation using knowledge-based techniques. IEEE Transactions on Medical Imaging. 1998; 17(2):187–201. [PubMed: 9688151]
58. Clark, MC.; Hall, LO.; Goldgof, DB.; Velthuizen, R.; Murtagh, R.; Silbiger, MS. Unsupervised brain tumor segmentation using knowledge-based fuzzy techniques. In: Teodorescu, HN.; Kandel, A.; Jain, LC., editors. Fuzzy and neuro-fuzzy systems in medicine. Lille: CRC; 1998. p. 137-169.
59. Clark M, Goldgof D, Hall LO, Clarke L, Silbiger M, Li C. MRI segmentation using fuzzy clustering techniques integrating knowledge. IEEE Engineering in Medicine & Biology. 1994; 13(5):730–742.
60. Pham DL, Prince JL. Adaptive fuzzy segmentation of magnetic resonance images. IEEE Transactions on Medical Imaging. 1999; 18(9):737–752. [PubMed: 10571379]
61. Bezdek JC, Hall LO, Clarke LP. Review of MR image segmentation techniques using pattern recognition. Medical Physics. 1993; 20:1033–1048. [PubMed: 8413011]
62. Pham DL, Prince JL. An adaptive fuzzy C-means algorithm for image segmentation in the presence of intensity inhomogeneities. Pattern Recognition Letters. 1999; 20(1):57–68.
63. Ahmed MN, Yamany SM, et al. A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data. IEEE Transactions on Medical Imaging. 2002; 21(3):193–199. [PubMed: 11989844]

64. Bensaid AM, Bezdek JC, Hall LO, Velthuizen RP, Clarke LP. Partially supervised fuzzy c-means algorithm for segmentation of MR images. Proc SPIE. 1992; 1710:522–528.

65. Bouix S, Martin-Fernandez M, Ungar L, Nakamura M, Koo MS, McCarley RW, et al. On evaluating brain tissue classifiers without a ground truth. NeuroImage. 2007; 36:1207–1224. [PubMed: 17532646]

Biographies

Prodip Hore received his B.Tech degree from the Institute of Engineering and Management, Kolkata, in 2002, and the masters and Ph.D degree in computer science from the University of South Florida, Tampa, in 2007. He is currently working as a analytic scientist in Fair Isaac Corporation, located in San Diego. His interests include machine learning, data mining, image processing, and optimization.

Lawrence O. Hall is a Professor and the Chair of the Department of Computer Science and Engineering at University of South Florida. He received his Ph.D. in Computer Science from the Florida State University in 1986 and a B.S. in Applied Mathematics from the Florida Institute of Technology in 1980. He is a fellow of the IEEE. His research interests lie in distributed machine learning, extreme data mining, bioinformatics, pattern recognition and integrating AI into image processing. The exploitation of imprecision with the use of fuzzy logic in pattern recognition, AI and learning is a research theme. He has authored or co-authored over 60 publications in journals, as well as many conference papers and book

chapters. Some recent publications appear in the IEEE Transactions on Pattern Analysis and Machine Intelligence, Neural Computation, Information Fusion, IEEE Transactions on Systems, Man, and Cybernetics, IEEE Transactions on Evolutionary Computing, the International Conference on Data Mining, the Multiple Classifier Systems Workshop, and the FUZZ-IEEE conference (http://isl.csee.usf.edu/ailab/hall.html).

He received the IEEE SMC Society Outstanding contribution award in 2000. He received an Outstanding Research achievement award from the Univ. of South Florida in 2004. A past president of NAFIPS. The former vice president for membership of the SMC society. He was the President of the IEEE Systems, Man and Cybernetics society for 2006–7. He was the Editor-In-Chief of the IEEE Transactions on Systems, Man and Cybernetics, Part B, 2002–05. Also, associate editor for IEEE Transactions on Fuzzy Systems, International Journal of Intelligent Data Analysis, the International Journal of Pattern Recognition and Artificial Intelligence and International Journal of Approximate Reasoning.

Dmitry B. Goldgof received his M.S. degree in Computer Engineering from the Rensselaer Polytechnic Institute in 1985 and the Ph.D. degree in Electrical Engineering from the University of Illinois at Urbana-Champaign in 1989. He is currently a Professor and an Associate Chair in the Department of Computer Science and Engineering and a member of H. Lee Moffitt Cancer Center and Research Institute, where during 2002–2003 he held a position of Professor in Bioinformatics and Cancer Control. Previously, Dr. Goldgof held visiting positions at the Department of Computer Science at the University of California at Santa Barbara and at the Department of Computer Science at University of Bern in Switzerland. Dr. Goldgof's research interests include image and video analysis, computer vision and pattern recognition, bioinformatics and bioengineering.

Dr. Goldgof has graduated 12 Ph.D. and 35 M.S. students, edited four books, published over 60 journals and over 130 conference papers. Professor Goldgof is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE) “for contributions to computer vision and biomedical applications”. Professor Goldgof has served as IEEE Distinguished Visitor 2004–2006 and on the Board of Governors of Systems, Man and Cybernetics Society in 2007. Dr. Goldgof was awarded Annual Pattern Recognition Society Awards in 1993 and 2002. One of his papers was selected by International Medical Informatics Association for 2000 Yearbook containing “the best of medical informatics.” Dr. Goldgof is an Associate Editor for IEEE Transactions on Systems, Man and Cybernetics B and for International Journal of Pattern Recognition and Artificial Intelligence. He has been involved in numerous professional society activities, has served as North American Editor for Image and Vision Computing Journal, member of the Editorial Board of the Pattern Recognition, and Associate Editor for IEEE Transactions on Image Processing.

Yuhua Gu received a Bachelor Degree in Applied Mathematics from Fudan University, Shanghai in 2001 and a M.S. in Department of Mathematics from University of South Florida in 2003. He is now a Ph.D. candidate in Department of Computer Science & Engineering from University of South Florida.

Andrew A. Maudsley is a Professor of Radiology at the University of Miami. He received a Ph.D. in Physics from Nottingham University, U.K., in 1976, during which time he worked on one of the first implementations of magnetic resonance imaging under the direction of Sir Peter Mansfield, who later received the Nobel Prize for his contributions to the field of MRI. Dr. Maudsley's research has concentrated on methodological development of MR spectroscopic imaging and application for clinical studies of the brain. He has co-authored over 100 journal publications, received an ISMRM gold medal award for contributions to the field of MRI, and is a member of the NIH study section for medical imaging.

Ammar Darkazanli is a Sr. Software Manager at Boston Scientific. He received his Ph.D. in Electrical and Computer Engineering from the University of Arizona in 1993 and a B.Sc./M.Sc. in Electrical Engineering and Applied Physics from Case Western Reserve University in 1988.

Ammar's interest has been developing software and building databases for the medical community. Early work included the development of pulse sequences for MRI and the use of MRI in guiding Ultrasound surgery. He also worked on Metabolite Image and Data Analysis System (MIDAS) and most recently has joined Boston Scientific to lead a team to support LATITUDE, a patient management system which communicates with Pulse Generators (pacemakers/ICD and CRT) implanted in the patients.

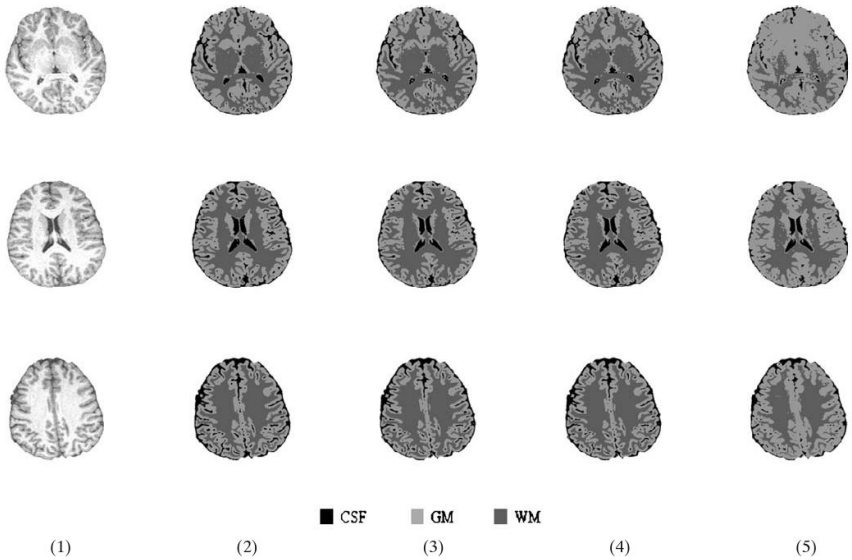

Figure 1.
1.5 Tesla: Volume MN018. Segmentation of slices 38 (*first row*), 48 (*second row*), and 58 (*third row*): (*1*) RAW T1 (*2*) FSL (*3*) SPFCM (*4*) OFCM (*5*) SPM.

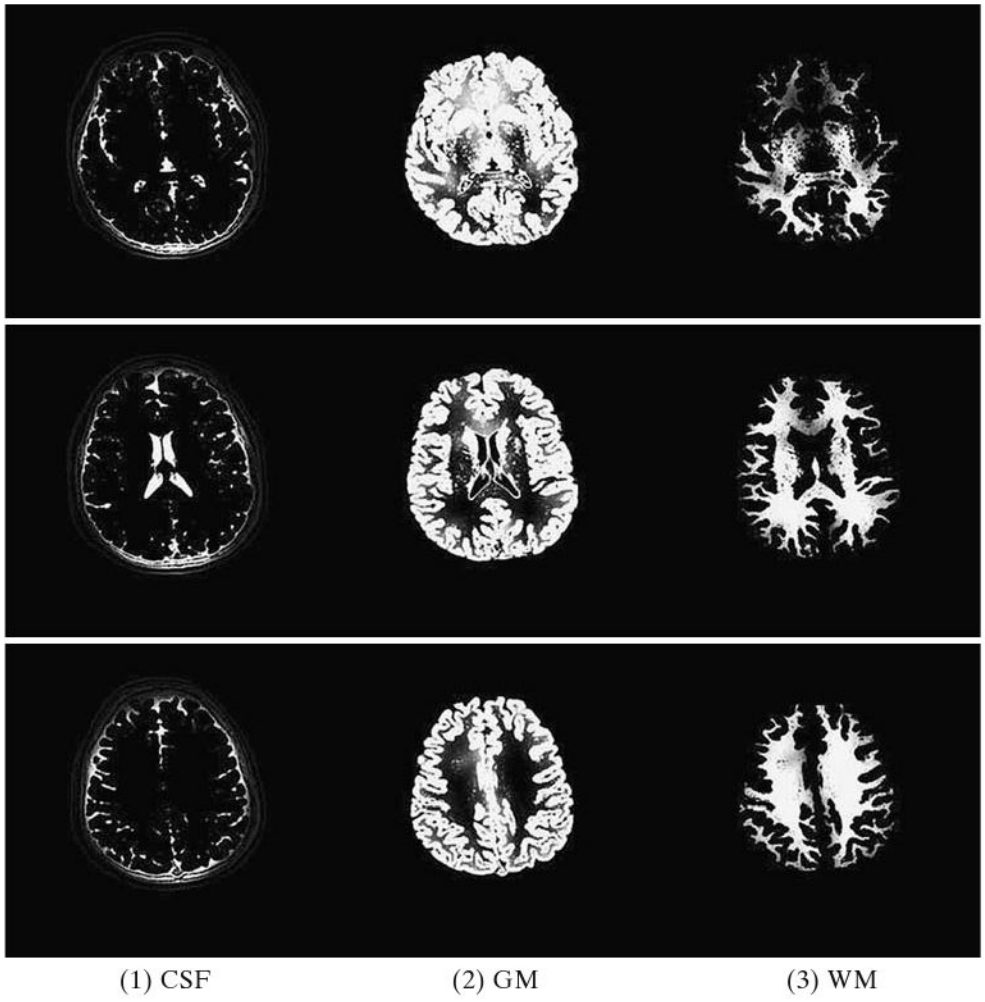

Figure 2.
1.5 Tesla: Volume MN018: original segmentation results with skull (native space) from SPM, slices 38 (*first row*), 48 (*second row*), and 58 (*third row*).

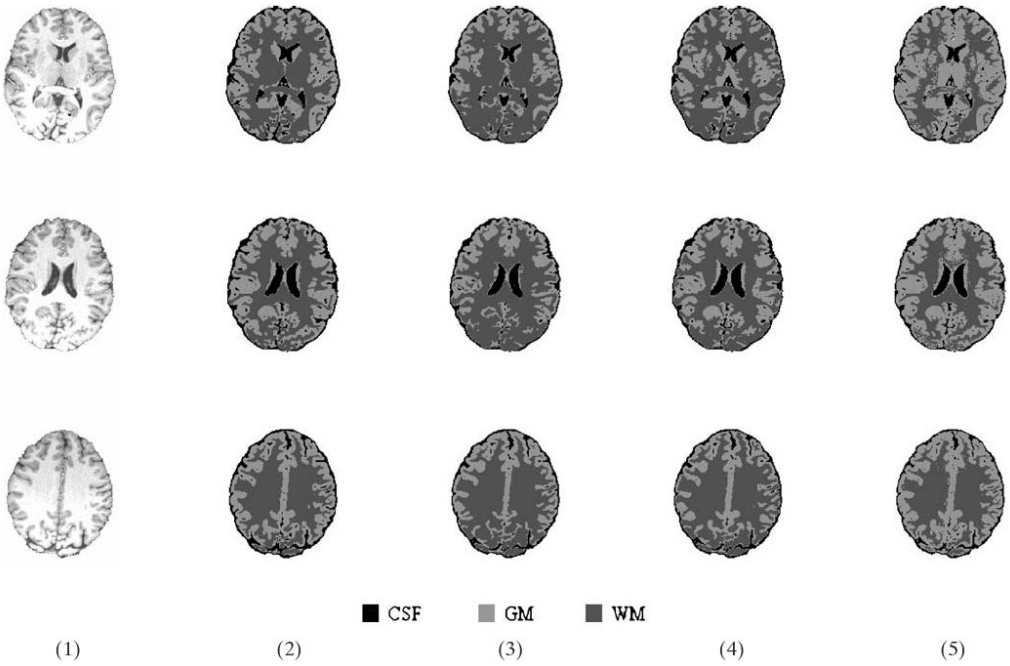

Figure 3.
3 Tesla: Volume Vol060. Segmentation of slices 70 (*first row*), 80 (*second row*), 90 (*third row*): (1) RAW T1 (2) FSL (3) SPFCM (4) OFCM (5) SPM.

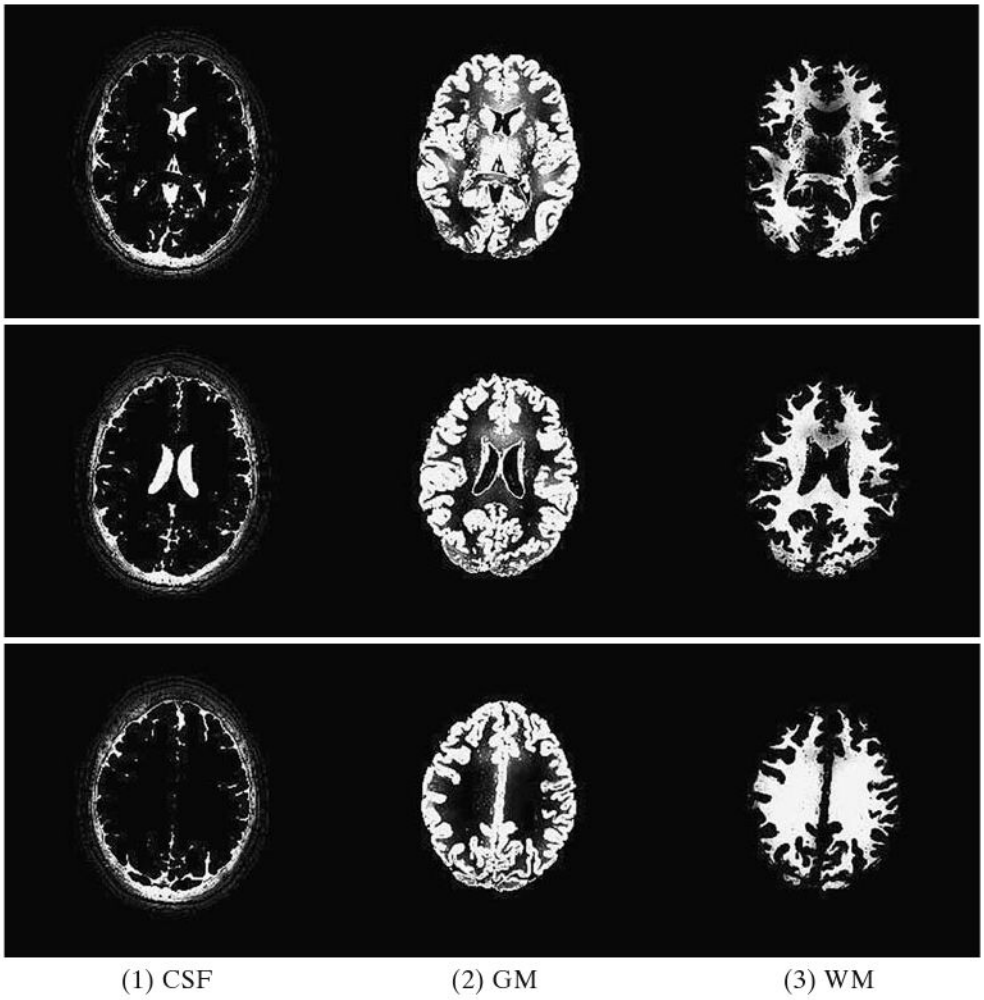

Figure 4.
3 Tesla: Volume Vol060: original segmentation results with skull (native space) from SPM, slices 70 (*first row*), 80 (*second row*), and 90 (*third row*).

Table 1

Image acquisition parameters.

Source	Modality	Slice thickness (mm)	TR (ms)	TE (ms)
UM 1.5T data	T1	1.5	1800	4.38
UM 3T data	T1	1	2150	4.38
UCSF 1.5T data	T1	1.5	1970	4.38

TR repetition time, *TE* echo time, *UM* University of Miami, *UCSF* University of California at San Francisco

Table 2

Average match and standard deviation in () of the segmentations of FSSP using SPFCM or OFCM, and SPM compared to FSL on the 1.5 T volumes.

	CSF (%)	GM (%)	WM (%)	Average (%)
SPFCM	84.05 (2.68)	84.10 (3.59)	97.47 (2.94)	88.54
OFCM	88.93 (2.16)	85.52 (2.95)	96.56 (3.59)	90.33
SPM	75.65 (12.87)	87.85 (5.84)	62.64 (5.46)	75.38

All values are expressed in percentage.

Table 3

Average match and standard deviation in () of the segmentation of FSSP using SPFCM or OFCM, and SPM compared to FSL on the 3 T volumes.

	CSF (%)	GM (%)	WM (%)	Average (%)
SPFCM	85.69 (1.83)	75.37 (2.44)	98.12 (0.71)	86.39
OFCM	87.30 (1.3)	77.94 (2.0)	97.35 (0.86)	87.53
SPM	83.49 (15.41)	81.08 (6.89)	59.81 (6.43)	74.79

All values are expressed in percentage.

Table 4

Average running time and standard deviation in (), in minutes, of the FSL, FSSP using SPFCM or OFCM, and SPM algorithms and their speed up compared to FSL.

	FSL (min)	SPFCM (min)	OFCM (min)	SPM (min)	SU_SPFCM	SU_OFCM	SU_SPM
1.5 T	22.53 (5.99)	1.30 (0.35)	2.09 (0.41)	20.50 (4.14)	17.33	10.77	1.09
3T	6.52 (0.56)	0.68 (0.07)	1.29 (0.19)	7.43 (0.86)	9.58	5.05	0.87

SU speed up

Table 5

SPFCM algorithm matching percentage with FSL on 1.5 T training volume MN011 with and without post processing.

Post processing	CSF	GM	WM	Avg
Off	78.08	83.43	99.22	86.91
On	84.68	84.28	99.28	89.41

Table 6

SPFCM algorithm on the 3 T training volume Vol009 matching percentage with FSL with and without post processing.

Post processing	CSF	GM	WM	Avg
Off	82.48	78.76	97.48	86.24
On	87.18	77.81	98.16	87.71

Table 7

OFCM algorithm on the 1.5 T training volume MN011 matching percentage with FSL with and without post processing.

Post processing	CSF	GM	WM	Avg
Off	78.08	83.43	99.22	86.91
On	86.72	84.85	99.18	90.25

Table 8

OFCM algorithm on the 3 T training volume Vol009 matching percentage with FSL with and without post processing.

Post processing	CSF	GM	WM	Avg
Off	83.78	80.10	96.92	86.93
On	87.38	79.84	97.42	88.21