INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Cloud Models: Their Construction and Employment in Automatic MRI Segmentation of the Brain

P. A. V. Miranda *A. X. Falcão*
J. K. Udupa

Technical Report - IC-10-08 - Relatório Técnico

March - 2010 - Março

Cloud Models: Their Construction and Employment in Automatic MRI Segmentation of the Brain

Paulo A. V. Miranda[*] Alexandre X. Falcão[†]

Jayaram K. Udupa, Fellow, IEEE[‡]

Abstract

A *cloud* is a triple consisting of a fuzzy object model, a delineation algorithm, and a criterion function for evaluating delineations. It employs recognition and delineation in a tightly coupled manner to accomplish image segmentation. It captures shape variations of a given object/object assembly to form an uncertainty region for its boundary. For any image position, delineation is executed in the uncertainty region to obtain a candidate object/object assembly, and the criterion function assigns a score to it. Image segmentation is defined by the candidate with the highest score. This work presents and compares three cloud models in automatic MR-T1 image segmentation of the cerebrum, the cerebellum, and cerebral hemispheres. These structures are connected in several parts, imposing serious segmentation challenges. The results show that the methods are fast, accurate, and can eliminate user intervention or, at least, reduce it to simple corrections. Their applications go beyond medical imaging to new vistas in various areas served by image segmentation.

1 Introduction

Image segmentation involves object *recognition* and *delineation* [23]. Recognition is the task of determining an object's approximate location in the image. Delineation completes segmentation by defining the exact spatial extent of the object. Humans usually outperform computers in object recognition, but the reverse is true for delineation. While the user can often solve the recognition problem by simple point (seed) selection or by an effective initialization action, precise delineation is challenging due to the intra and inter operator subjectivity. On the other hand, computers can be very precise, even when they are not accurate, but the absence of global information (e.g., an object model) makes object recognition a difficult task for them. This explains why some successful interactive approaches combine recognition by the user with delineation by the computer in a synergistic way, for effective and foolproof segmentation [8, 9, 3].

[*] University of Campinas, Institute of Computing, Campinas, SP, Brazil, e-mail: pavm@ic.unicamp.br

[†] University of Campinas, Institute of Computing, Campinas, SP, Brazil, e-mail: afalcao@ic.unicamp.br

[‡] Department of Radiology, Univ. of Pennsylvania, Philadelphia, PA, USA, e-mail: jay@mail.med.upenn.edu

Segmentation methods can be roughly divided into *model-based* and *image-based* approaches. Model-based methods create statistical models by employing supervised learning. A training set of object's instances is provided with appropriate human interaction and these data are registered into a common reference space to form the model. Active shape models [5] (ASM) and atlas-based approaches [12, 25, 24] are examples of model-based methods that have been used for MR-image segmentation of anatomic structures in the brain [7, 13]. Accurate registration is a separate problem in these methods which is also required during segmentation. In ASM, landmarks have to be selected on the surface of the training objects and their correspondence provides a statistical model of possible variations in shape. The registration between the image and the model during segmentation also sometimes ignores important image information, by the act of forcing the results to fit with the model. Brain atlases are usually created by registration of training images based on certain landmarks (e.g., anterior and posterior comissures) and deformation fields. In the reference space, image structures suffer from different degrees of distortion, making the matching among corresponding voxels inexact. Image-based methods in turn exploit image properties for more effective delineation, but their lapses in global information makes object recognition an insurmountable problem.

In view of these dilemmas, some recent methods have addressed automatic segmentation by combining model-based approaches for recognition with image-based approaches for delineation [17, 4, 21, 22]. Essentially, the model plays the role of the human operator while an image-based algorithm performs delineation, and both operate in a synergistic way until an optimum state is reached. In this paper, we pursue our previous work on *object cloud models* [21, 22] which present the following advantages: the cloud models dismiss registration during training and segmentation; they take into account the entire object's boundary during delineation and recognition rather than only some *control points*, as in [5]; and they can be easily extended to multidimensional images.

An *object cloud model* (OCM) was introduced in [21] as a triple comprising a fuzzy object, a delineation algorithm, and a criterion function. It captures shape variations of a given object to form an uncertainty region for its boundary. For any image position, delineation is executed in the uncertainty region to obtain a candidate object and the criterion function assigns a score to it. Image segmentation is defined by the candidate with the highest score. In order to capture more shape differences, we proposed the *cloud bank model* (CBM) in [22], which uses multiple clouds per object. In some applications, however, multiple objects may define a *cloud system* by adding their relative position into the model. This is the case of MR-T1 images of the brain and several other medical imaging applications. Therefore, we propose in this paper the *cloud system model* (CSM) to handle multiple objects simultaneously. The methods are compared for the automatic MR-T1 image segmentation of: (S1) the brain (without stem), (S2) the cerebral hemispheres, (S3) the cerebellum, and (S4 and S5) the right and left hemispheres.

In the literature, most approaches to segment S1-S5 are based on atlas registration or surface-extraction techniques. CSM does not require registration and segments S1-S5 simultaneously, precisely and accurately. Besides, a quick evaluation of several other tools revealed that CSM is much simpler and faster. For example, it is about 5 times faster than BrainVisa [19], with no need for manual landmark specification. As compared with the recent work in [16], which takes about 3 hours using deformable registration and 17 atlases, the proposed approach is much simpler, faster and produces similar Dice measures for S3. Other examples are SurfRelax [14], which takes about 6 minutes and requires image warping onto a template, CLASP[1] and FreeSurfer [6], which take hours to complete segmentation of S1-S5.

We present the general definitions of all object cloud models - OCM, CBM, and CSM - in Section 2; instantiate them for brain MR-T1 image segmentation in Section 3; discuss experimental results in Section 4; and state conclusions in Section 5. The experiments show the advantages of CSM and improvements over our previous works [21, 22] for this particular application.

2 Object Cloud Models

For a given object of interest (e.g., brain, cerebellum), a set of training images with the object's instances must be provided. These instances should capture among them shape variations of that object in order to teach the computer how to recognize it in the image. In atlas-based approaches [12], the images are registered to a chosen reference image by finding a geometric transformation that best matches them according to a criterion of similarity. During the registration process, the reference image remains fixed while the others are deformed into its geometric space. Following this, the model is obtained as the averaged image template together with the tissue distribution maps obtained by averaging segmentations over all subjects. Suitable reference selection is itself a problem, since the atlas can be biased towards the anatomy of the chosen image. Hence, this image should be the one that best represents the anatomy of a population under study [25].

Suppose, instead of registering the training instances, we only translate their binary segmentations on to a common reference point (geometric center) and compute their average. This results in an image with a fuzzy appearance that resembles a *cloud* (Figure 1a). From it we may obtain relevant shape informations such as prior boundary knowledge (see Figure 1b and Section 2.3). Each *cloud* image also defines (i) an interior region consisting of voxels that belong to all training instances, (ii) an exterior region with voxels that do not belong to any instance, and (iii) an uncertainty region composed of voxels that belong to some but not to all instances (Figure 1c).

The *cloud* model (OCM) is a triple that consists of a fuzzy object (*cloud* image), a delineation algorithm A (this may be any algorithm, but, for the reasons mentioned earlier, preferably an image-based approach), and a functional F. To segment a new image, the *cloud* moves over the image and, for each position, algorithm A is executed inside the uncertainty region to obtain a candidate segmentation. The functional is evaluated on this segmentation to obtain a matching score for recognition, by taking into account local and global object properties (e.g., shape and texture). The desired segmentation is expected to be the one with maximum score [21] (Figures 2a-c). However, when the variability in the training set is too high the shape information within the *cloud* image tends to degenerate and the method loses precision. To circumvent this problem, we may separate the training

[1] URL: http://www.bic.mni.mcgill.ca/

instances into *groups* (clusters) of high similarity in shape, rotation and size (texture may be used as well). Each group defines its own *cloud* image resulting in a bank of clouds, and the desired segmentation is expected to be the one with maximum score among segmentations obtained from all individual *clouds* from this *cloud bank model* (CBM) [22]. In this manner, the need to define a reference image is completely obviated. As new images are added to the training set, they start new groups or are inserted in some existing group (the groups may have overlap). In medical imaging and other applications (e.g., license plate recognition), it is possible to acquire images as per a disciplined regimen so that a small number of groups in the bank will suffice. Preprocessing, that depends only on the image being processed, can also help in reducing the number of groups.

Note that delineation is constrained in the uncertainty region, and it also exploits prior shape information (see Section 2.3) which are defined by the model. Recognition is based on the functional, but it is applied to the delineated objects. Thus, the model employs recognition and delineation in a tightly coupled manner [17].

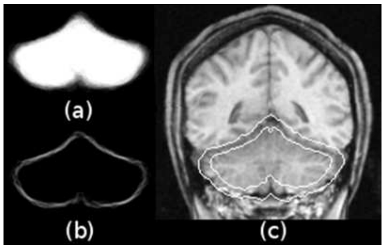

Figure 1: (a) A coronal slice of the 3D cloud image of the cerebellum. (b) The shape-based weight image. (c) Example of an uncertainty region over a slice of a test image.

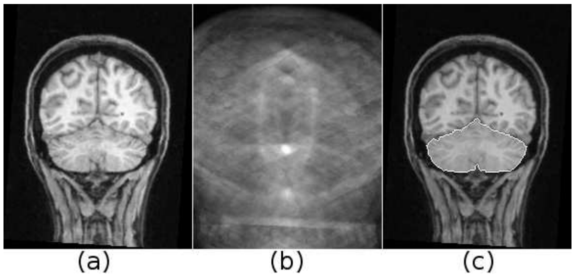

Figure 2: (a) An input image. (b) The recognition score for all positions. (c) The final segmentation at the best location.

In the case of multiple objects, each object has its own OCM (or CBM), separately, and the segmentation can follow independently, or in a hierarchical search. In this last case, the search spaces for the internal objects are constrained by the larger objects that subsume them (e.g., Figure 3, used in [21, 22]). However, in some applications (such as medical), the arrangement among objects does not change, so it is possible to make better

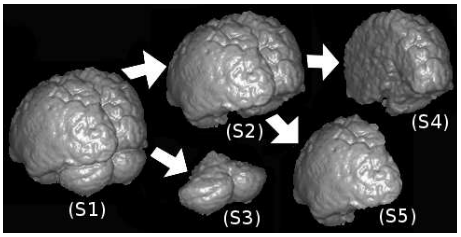

Figure 3: Hierarchical search for the brain structures S1-S5: S1 is the brain without the brain stem, S2 represents the cerebrum, S3 is the cerebellum, and S4 and S5 are the right and left cerebral hemispheres.

use of contextual information, by computing their mean relative positions with respect to a common reference point (centroid of all objects). As a result, we have a *cloud system* (Figure 4a). To segment a new image, we consider the prior displacement knowledge in order to fix the position of each object cloud relative to the moving reference point. For each search position, delineation is done inside the uncertainty regions of all object clouds, and a combined score for recognition is obtained by the functional F. This simultaneous treatment of multiple objects makes it less likely to miss the right position, due to the better use of contextual information (Figures 4b-c) [4].

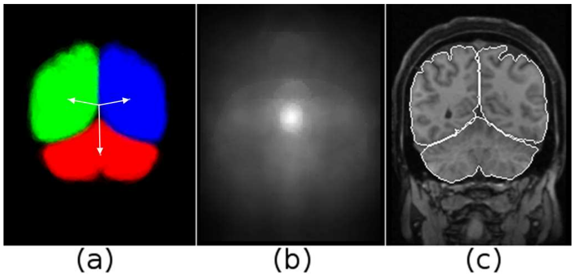

Figure 4: (a) A cloud system for $c = 3$ objects shown in RGB color space. (b) The recognition score for all positions and (c) the final segmentation handling multiple objects simultaneously.

Again, when the variability of the training instances is too large, we can separate the instances into *groups* of high similarity, but now we must also consider possible correlations among the objects in terms of relative position, rotation, and size. This leads to a *cloud system model* (CSM), which is formally presented below. The previous cloud models (OCM,CBM) are subsumed by CSM as particular cases.

An image $\hat{I}$ is a pair $(\mathcal{I}, \vec{I})$ where $\mathcal{I} \subset Z^n$ is the image domain and $\vec{I}(p)$ assigns a set of b scalars $I_i(p)$, $i = 1, 2, \ldots, b$, to each voxel $p \in \mathcal{I}$. This definition applies to multi-dimensional and multi-parametric images. We are interested in $n = 3$ and $b \geq 1$. The index i is removed when $b = 1$. For c objects $l = 1, 2, \ldots, c$, a segmentation instance is represented by a labeled image $\hat{L} = (\mathcal{I}, L)$, in which each label $1 \leq L(p) \leq c$ assigns a voxel $p \in \mathcal{I}$ to one object out of c objects, and $L(p) = 0$ is used to designate background voxels. A binary image $\hat{B} = (\mathcal{I}, B)$ may be used to represent each object such that $B(p) = 1$ for object voxels and $B(p) = 0$ for background voxels.

Training instances are given as a set of labeled images $\hat{L}_i = (\mathcal{I}_i, L_i)$, $i = 1, 2, \ldots, N$. Suppose these instances are separated into m groups of high similarity, as to be discussed in Section 2.1. Each group is represented by a set $\mathcal{G}_g$ of images, $g = 1, 2, \ldots, m$, such that $\hat{L}_i \in \mathcal{G}_g$ if the image $\hat{L}_i$ is in the gth group. Let $\hat{B}_{i,l} = (\mathcal{I}_i, B_{i,l})$ be the binary image of the lth object in the ith image (i.e., $B_{i,l}(p) = 1$ if $L_i(p) = l$, and $B_{i,l}(p) = 0$ otherwise). For any given object label l and group g, the average of the binary images $\hat{B}_{i,l}$ for all i in $\mathcal{G}_g$, after translating them to a fixed reference point, creates a *cloud image* $\hat{C}_{g,l} = (\mathcal{C}, C_{g,l})$, where $C_{g,l}(p) \in [0, 1]$. For any cloud, $C_{g,l}(p) = 1$ in its interior, $0 < C_{g,l}(p) < 1$ in its uncertainty region, and $C_{g,l}(p) = 0$ in its exterior. In the single object case (i.e., $c = 1$), the method becomes the same as in CBM [22], and it becomes a single OCM [21] when $c = 1$ and $m = 1$. But in the case of multiple objects (i.e., $c > 1$), it also exploits the relative positions among the objects within each group. For a given group g, the arrangement of the object clouds is captured by the displacement vectors $\vec{D}_{g,l}$, $l = 1, 2, \ldots, c$, which store the average positions of the object's centroids in relation to their joint centroid for all images in $\mathcal{G}_g$.

$$\vec{D}_{g,l} = \frac{1}{|\mathcal{G}_g|} \sum_{\forall i | \hat{L}_i \in \mathcal{G}_g} \left(\vec{P}_{i,l} - \vec{P}_i \right), \tag{1}$$

where $|\mathcal{G}_g|$ is the cardinality of the set $\mathcal{G}_g$, $\vec{P}_{i,l}$ is the centroid's coordinates of the object l in the binary image $\hat{B}_{i,l}$, $l = 1, 2, \ldots, c$, and $\vec{P}_i$ is the centroid's coordinates of their union in the label image $\hat{L}_i$. Note that this formulation also includes the single object case since $\vec{D}_{g,l}$ becomes the null vector when $c = 1$.

The following subsections provide more details about all relevant parts of the CSM, such as the grouping strategy, the object search, the graph assembly, the delineation algorithm A, and the functional F. The model components and parameters are then customized for the specific application of MR-T1 image segmentation of the brain as described in Section 3.

2.1 Grouping

The grouping can be done by representing the training instances $\hat{L}_i$ as nodes of a complete graph; the arcs between $\hat{L}_i$ and $\hat{L}_j$ are weighted by a metric that valuates their similarity. In the single object case ($c = 1$), we may consider the Dice similarity as this metric after centralizing the training instances by their centroid vectors $\vec{P}_{i,l}$ (Figure 5a). In the case of multiple objects ($c > 1$), we centralize the label images $\hat{L}_i$, $i = 1, 2, \ldots, N$, by $\vec{P}_i$ and consider a combined similarity value (e.g., the mean Dice similarity among corresponding object's instances). The groups are then selected as maximal cliques [26], wherein all pairs of training instances have similarity above a threshold. Thresholding helps us to assure that only compatible data are used within the same cloud system, and the size of the uncertainty regions can be controlled by the threshold (Figure 5b). On the other hand, we must keep the number of groups as low as possible in order to make the method computationally efficient.

We start with N maximal cliques, such that each contains one training image. Since these cliques may have overlap, we select only a minimal subset of them that comprises all the images. In most cases this can be accomplished by successively applying three simple rules: eliminate repeated cliques (or cliques having all unselected nodes contained in some other clique), select all cliques with at least one exclusive element, and ignore remaining cliques that have all elements already selected. When this fails, there is a cycle of dependencies that may be broken by arbitrarily selecting one of the remaining cliques, and then resuming the three rules.

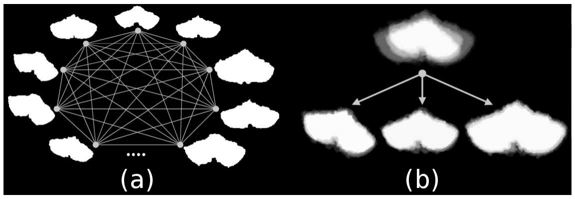

Figure 5: (a) The training instances ($c = 1$) are mapped as nodes of a complete graph, (b) the groups are selected as maximal cliques.

If there are only a few training images available, they may be complemented with images created by random transformations (within acceptable limits) of the given images which capture variations that are likely to happen in the application.

2.2 Locating objects

Let $\hat{I} = (\mathcal{I}, I)$ be an image to be segmented. For each search position with coordinates given by $\vec{p}$, each cloud image $\hat{C}_{g,l}$ is positioned with its center at $\vec{q} = \vec{p} + \vec{D}_{g,l}$, and its uncertainty region is projected over a set of voxels $\mathcal{U} \subset \mathcal{I}$. That is, the displacement vectors $\vec{D}_{g,l}$ are used to fix the cloud's position for each object relative to the moving search point $\vec{p}$. Then, for each position $\vec{p}$ of a search region, a score F_l is obtained for each label $l = 1, 2, \ldots, c$, by analyzing the candidate segmentations computed by the delineation algorithm for each cloud in a given cloud system. A combined score for recognition is obtained by the functional F, such as the mean value $\left(\sum_{l=1}^{c} F_l\right)/c$. This process is repeated for all groups, and the best location among all groups is selected as the final result.

A multiscale search can be used to speed up this recognition task, by starting the search at the lowest resolution and refining the best detected locations in the higher resolutions. During this refinement in the higher resolutions, we may also allow each cloud of a cloud system to move independently from the others in order to have more flexibility.

2.3 Arc-weight estimation for delineation

Image-based delineation algorithms usually make direct/indirect use of some image-graph concept, and their success strongly depends on a suitable arc-weight estimation, which usually takes into account image attributes and/or object information obtained by supervised learning [23]. For the given image $\hat{I} = (\mathcal{I}, I)$ to be segmented, we associate a weight image

$\hat{W} = (\mathcal{I}, W)$ which is in turn associated with a graph $(\mathcal{I}, \mathcal{A})$. The graph's nodes are the voxels $p \in \mathcal{I}$ and arcs $(p, q) \in \mathcal{A}$ are defined between 6-neighbors. For convenience, we store the weights in voxel resolution and use their interpolated values during execution for the graph. That is, each arc $(p, q) \in \mathcal{A}$ is weighted by the mean value $w(p, q) = \frac{W(p)+W(q)}{2}$. The weight $W(p)$ assigned to each voxel $p \in \mathcal{I}$ is a linear combination of an image-based weight $W_i(p)$, an object-based weight $W_o(p)$, and a shape-based weight $W_s(p)$ provided by the cloud model.

$$W(p) \quad = \quad \lambda_i W_i(p) + \lambda_o W_o(p) + \lambda_s W_s(p), \tag{2}$$

where $\lambda_i + \lambda_o + \lambda_s = 1$. The weight $W_i(p)$ aims at capturing discontinuities that may exist between homogeneous regions and is taken as the magnitude of an image gradient. The weights $W_o(p)$ take into account prior knowledge about the intensities of the objects under consideration in order to characterize the discontinuities that exist between them and the rest of the image. This weight is usually application-dependent (see Section 3.2), although some general techniques exist to estimate them [23]. The weight $W_s(p)$ encodes prior shape information obtained from each *cloud* image $\hat{C}_{g,l}$ (Figure 1a) by computing its gradient magnitude for all groups and labels (Figure 1b). It is combined with the other weights as each *cloud* moves over the image (i.e., each cloud has its own $W_s(p)$), usually with a low λ_s value since its major role is only to discriminate regions with poorly defined borders.

2.4 Delineation algorithm

For delineation, we use an algorithm called IFT-SC (*IFT segmentation by Seed Competition*) which is based on the image foresting transform [10] (IFT) and is supported by the theoretical foundations given in [20].

For each search position, each cloud image $\hat{C}_{g,l}$ defines an uncertainty region as a set $\mathcal{U} \subset \mathcal{I}$ of voxels (Figure 1c). The interior and exterior regions contain boundary voxels, which have at least one voxel in $\mathcal{U}$ as a 6-neighbor. These boundary voxels form one internal set $\mathcal{S}_i$ and one external set $\mathcal{S}_e$ of seeds for the IFT-SC. A *path* π in the image graph $(\mathcal{I}, \mathcal{A})$ is a sequence of adjacent voxels $\langle p_1, p_2, \ldots, p_n \rangle$. For the given set of seeds $\mathcal{S} = \mathcal{S}_i \cup \mathcal{S}_e$, the cost of a path π is defined by a *path-cost function.* The considered cost functions are:

$$f_1(\pi) \quad = \quad \begin{cases} \max_{i=1,2,\ldots,n-1} w(p_i, p_{i+1}) & \text{if } p_1 \in \mathcal{S} \\ +\infty & \text{otherwise} \end{cases} \tag{3}$$

$$f_2(\pi) \quad = \quad \begin{cases} \sum_{i=1,2,\ldots,n-1} [w(p_i, p_{i+1})]^{\eta} & \text{if } p_1 \in \mathcal{S} \\ +\infty & \text{otherwise} \end{cases} \tag{4}$$

When $\mathcal{U}$ contains the object's boundary (Figure 1c), it is expected that the arc weights within $\mathcal{U}$ are higher on the object's boundary than inside and outside it. The seed sets $\mathcal{S}_i$ and $\mathcal{S}_e$ compete for voxels in $\mathcal{U}$, such that a voxel receives label $L(p) = 0$ if the minimum-cost path comes from $\mathcal{S}_e$, and label $L(p) = l$ otherwise. The object is then defined as the union between the interior of the *cloud* and the voxels with labels $L(p) = l$ in $\mathcal{U}$.

The IFT solves this minimization problem by computing an *optimum-path forest* — a function P which contains no cycles and assigns to each node $q \in \mathcal{I}$ either its predecessor node $P(q) \in \mathcal{I}$ in the optimum path with terminus q or a distinctive marker $P(q) = nil \notin \mathcal{I}$, when $\langle q \rangle$ is optimum (i.e., q is said to be a *root* of the forest). The cost functions by Eq. 3 and 4 force the roots to be in $\mathcal{S}$ (Figure 6). As we change the parameter $\eta > 0$ in f_2, we obtain a whole family of solutions that includes f_1 as a special case in the limit when $\eta \to \infty$. This is noticeable from the empirical results shown in [22]. We note that, f_2 usually imposes more regularization to the object's boundary than f_1, but on the other hand, f_1 fits to the protrusions and indentations of the boundary better than f_2. The IFT-SC is especially important, because our model is being projected to allow user's intervention for small corrections if necessary, and it was shown in [20] that the boundaries obtained by the IFT-SC with path-cost function f_1 are also piecewise optimal. This property is essential to conserve user control during interactive corrections as discussed in [20], and these corrections can be quickly performed by using the DIFT algorithm [9].

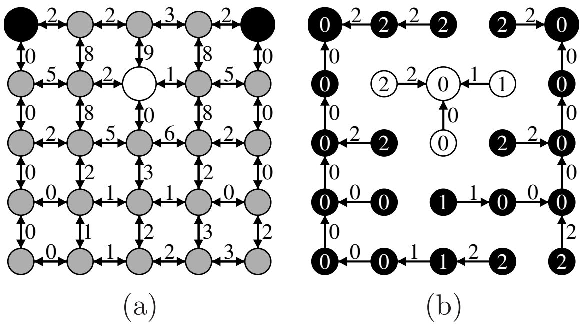

Figure 6: IFT-SC example: (a) A 4-neighborhood graph, where the numbers indicate the arc weights. Three seeds are selected, one is internal (white dot) and two are external (black dots). (b) An optimum-path forest for the path-cost function f_1. The numbers inside the nodes indicate the costs of the optimum paths, which are stored in a predecessor map P. The label $L(q) = 0$ (black), or $L(q) = l$ (white) of each seed q is propagated to all pixels within its respective optimum-path tree.

2.5 Functional for recognition

The functional F should be the one that best discriminates the objects under consideration, being consequently application-dependent. Several delineation algorithms are already based on some sort of functional, like for example a graph-cut measure [3]. Indeed, the IFT-SC with path-cost function f_1 also optimizes a graph-cut measure according to [20]. A question may be raised concerning the use of the same functional for both delineation and recognition score. This is usually not the best option, because the functionals for delineation are in general designed to best match boundary properties, while the recognition functional could be far more complex, taking into account more global information, such as the shape details and internal information. The functional used for brain segmentation is discussed in Section 3.3.

3 Application to brain segmentation

In this work, we consider the segmentation of the brain structures S1-S5 as summarized in Figure 3. The segmentation of the left and right cerebral hemispheres enables the investigation of asymmetries in shape and texture which may be related to several degenerative diseases. The separation of the cerebellum makes its analysis possible independently of the rest of the brain, which is important in understanding the relationships between aging and the decline in cognitive functioning [16]. The hemispheres are connected through the corpus callosum. The cerebellum is connected to the rest of the brain through the brain stem and through its top due to partial volume. The absence of a clear boundary between these structures poses a challenge for segmentation (Figure 7).

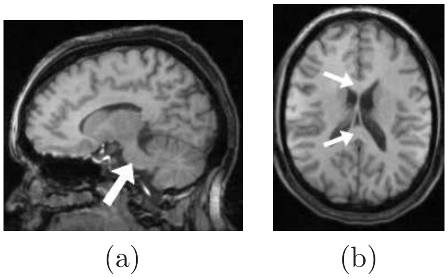

Figure 7: (a) The cerebellum is connected to the cerebrum through the brain stem. (b) The cerebral hemispheres are connected through the corpus callosum.

This application of MR brain image segmentation perfectly fits the requirements for the CSM model. The images are acquired in a well controlled environment, where the patient remains lying in a horizontal position without moving. In extreme cases (e.g., Claustrophobia) even mild sedation may be adopted. Hence, only a small number of clouds will suffice as desired.

3.1 Preprocessing & grouping

As stated in Section 2, preprocessing that depends only on the image being processed can help in reducing the total number of groups required. The MR-T1 images are interpolated to the same cubic dimensions ($0.98mm^3$) and aligned by the mid-sagittal plane [2] (MSP). This approach is fast (a few seconds), free of parameters, and independent of templates. Interpolation and alignment reduces the number of groups by reducing the data variability, and the MSP also reduces the search region. Note that, the MSP alignment operation does not depend on any reference image. For grouping, we used the methodology of Section 2.1, with the combined similarity threshold of 0.8.

3.2 Arc-weight estimation

In this application, we may take advantage of the fact that most of the brain structures are surrounded by CSF in order to emphasize their boundaries. However, owing to the intensity non-standardness among images, and also the inhomogeneity, the partial volume effects, and their interplay [18], it is difficult to characterize the exact intensity profile of the CSF in advance. But we may still enforce a more specific intensity interval which is more likely to contain the transition from CSF to GM. Consider t_1 as a lower bound of the CSF intensity, and t_2 as a lower bound for the white matter. We can accentuate the intensity slope in the interval $[t_1, t_2]$ by considering the filtered image $\hat{I}' = (\mathcal{I}, I')$ as follows.

$$I'(p) = \begin{cases} I(p) & \text{if } I(p) < t_1 \\ (1-\gamma)t_1 + \gamma \cdot I(p) & \text{if } I(p) \in [t_1, t_2] \\ (t_2 - t_1)(\gamma - 1) + I(p) & \text{otherwise} \end{cases} \tag{5}$$

where $\gamma > 1$ defines the new slope value (e.g., $\gamma = 5$), such that differences within $[t_1, t_2]$ in the original image $\hat{I}$ become higher in $\hat{I}'$. Subsequently, we consider a simple application-specific weight $W_o(p)$ that emphasizes dark voxels close to bright areas in the filtered image $\hat{I}'$. For all voxels $p \in \mathcal{I}$, $W_o(p)$ is computed as the sum of differences $I'(q) - I'(p)$ for all brighter voxels q (i.e., $I'(q) > I'(p)$) in a small neighbourhood around p. Thus, a suitable emphasis is obtained in the superior part of the cerebellum, and the transitions from GM to WM become weakened (Figure 8). In the experiments we considered t_1 as the Otsu's optimal threshold (which gave a good lower bound for the CSF in all cases), and we considered t_2 as the mean intensity in $\hat{I}$ considering only values above t_1. The other weights $W_i(p)$ and $W_s(p)$ are computed as discussed in Section 2.3, $W_s(p)$ being especially important to complete poorly defined borders (Figure 7).

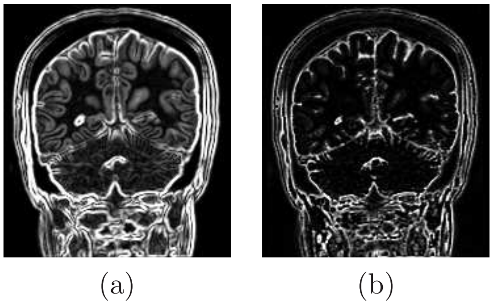
(a) (b)

Figure 8: (a) The image-based weight W_i, (b) the object-based weight W_o.

3.3 Recognition functional & delineation algorithm

The previous versions [21, 22] used the mean-cut measure [27] as the functional F_l, since it provides a good summary of the object's boundaries, being free of any undesirable bias. In this work, we improve on this functional by including a penalty factor based on our application-specific knowledge. The likelihood of having image intensities below t_1 inside the brain structures is supposed to be very low. Therefore, if the proportion of voxels below t_1 is high, among all voxels achieved for the object from $\mathcal{U}$, then this is a strong indication that the cloud is poorly positioned.

With respect to the delineation algorithm, some works [15] propose the use of a graph-cut algorithm with an increasing transformation (i.e., the exponential function). But since we have a potent arc-weight estimation strategy which also includes prior shape information (filling any gaps on the object's boundary), a graph-cut algorithm under this scenario will theoretically lead to an approximation of the IFT-SC according to [20]. Hence, it is preferable to use the faster IFT-SC algorithm.

Algorithm 1 below performs the IFT-SC delineation and functional computation F_l simultaneously for any given image location. It takes time proportional to the number of voxels in $\mathcal{U}$ (sublinear), when the priority queue Q is implemented properly [8]. For multiple objects, the scores F_l are combined as $F = (\sum_{l=1}^{c} F_l)/c$, as discussed in Section 2.2.

Algorithm 1 – Delineation algorithm for f_1

Input: Image $\hat{I}$, weight image $\hat{W}$, adjacency $\mathcal{A}$, seed sets $\mathcal{S}_i$ and $\mathcal{S}_e$, and uncertainty region $\mathcal{U}$.

Output: Label map L initially zeroed, an optimum-path forest stored in a predecessor map P initially with *nil* and functional $F_l = 0$ initially.

Auxiliary: Cost map c initially zeroed, variables cst, object acquisitions $oa = 0$, penalty acquisitions $pa = 0$ and cut size $sz = 0$ initially, priority queue Q initially empty, and status map s to indicate when a voxel has been inserted in Q (1), has never been inserted in Q (0), or has been removed from Q (2).

```
1.  For all $p \in \mathcal{U}$, set $c(p) \leftarrow +\infty$ and $s(p) \leftarrow 0$.
2.  For all $p \in \mathcal{S}_i$, set $L(p) \leftarrow l$, $s(p) \leftarrow 1$, and insert $p$ in $Q$.
3.  For all $p \in \mathcal{S}_e$, set $L(p) \leftarrow 0$, $s(p) \leftarrow 1$, and insert $p$ in $Q$.
4.  While $Q$ is not empty, do
5.      Remove from $Q$ a voxel $p$ such that $c(p)$ is minimum.
6.      Set $s(p) \leftarrow 2$.
7.      For each $q$ such that $(p,q) \in \mathcal{A}$, do
8.          If $c(q) > c(p)$, then
9.              Compute $cst \leftarrow \max\{c(p), \frac{W(p)+W(q)}{2}\}$.
10.             If $cst < c(q)$, then
11.                 If $s(q) = 1$, remove $q$ from $Q$.
12.                 Set $c(q) \leftarrow cst$, $L(q) \leftarrow L(p)$.
13.                 Set $P(q) \leftarrow p$.
14.                 Insert $q$ in $Q$ and $s(q) \leftarrow 1$.
15.         Else
16.             If $s(q) = 2$ and $L(q) \neq L(p)$, then
17.                 Set $F_l \leftarrow F_l + \frac{W(p)+W(q)}{2}$.
18.                 Set $sz \leftarrow sz + 1$.
19.     If $L(p) = l$, then
20.         Set $oa \leftarrow oa + 1$.
21.         If $I(p) < t_1$, then set $pa \leftarrow pa + 1$.
22. Set $F_l \leftarrow (F_l/sz) \times (1 - pa/oa)$.
```

Lines 1–3 initialize maps and insert seed voxels in Q. Lines 4–14 compute the maps L, c and s during the IFT. The main loop computes an optimum path cost from the seeds to every node p in a non-decreasing order of values (Lines 4–14). At each iteration, a path of minimum cost $c(p)$ is obtained in P when we remove its last voxel p from Q (Line 5). Lines 7–14 evaluate if the path that reaches an adjacent voxel q through p is cheaper than the current path with terminus q and update Q, $c(q)$, $s(q)$, $L(q)$ and $P(q)$ accordingly. The remaining lines compute the functional F_l on-the-fly.

Note that, before changing position of $\mathcal{U}$, the maps and auxiliary variables can be reinitialized in sublinear time, such that the search for a desired object can be done more efficiently. The above algorithm can be easily modified for f_2 if we substiture line 9 by $cst \leftarrow c(p) + \left[\frac{W(p)+W(q)}{2}\right]^{\eta}$.

3.4 Multiple object search

Instead of making a hierarchical search (i.e., the pipeline of Figure 3) as in [21, 22], we made the search simultaneously for the three objects S3, S4 and S5, as described in Section 2.2. Later, object S2 is obtained as S4 ∪ S5, and S1 as S3 ∪ S4 ∪ S5. The search was constrained only inside the MSP, and we also used a multiscale strategy to speed up the recognition task by using a three-level Gaussian pyramid.

4 Results

We have first evaluated the method on the MRI datasets of 40 normal subjects from both genders, in the age range from 16 to 49 years. The images were acquired with a 2T Elscint scanner and voxel size of $0.98 \times 0.98 \times 1.00$ mm^3. We used the leave-one-out approach to compute the mean and standard deviation of the Dice similarity measure between the ground truth and the automatic segmentation (Figures 10 and 9). Table 1 shows the results for different methods. The CSM model was executed with four groups, using path-cost function f_1, and with arc weights computed using $\lambda_i = 0.15$, $\lambda_o = 0.75$, and $\lambda_s = 0.10$. The obtained results are considerably superior to those of OCM [21], especially for the cerebellum; and are also much more precise (i.e., lower standard deviation) than the previous CBM [22]. The simultaneous search of multiple objects is also faster than the hierarchical search used in [21, 22]. The mean execution time per group using a 3GHz Pentium IV PC was 27.5 sec, while the hierarchical search took 41 sec. Therefore, the total mean execution time per data set to segment all objects is 110 sec. We also note that our results for brain segmentation S1 are considerably superior to those reported in [1] for *tree pruning* [1] and SPM2 [11] (mean errors of 9.4% and 14.1%, respectively), the latter being a widely used template-based approach for medical research. This is especially notable concerning the removal of the brain stem.

We also conducted experiments with 40 patient images which were acquired post-surgery, some of which had strong morphological changes (Figure 11). In this case, some special care has been taken. After training on the 40 controls, we considered a greater degree of contingency in the size of $\mathcal{U}$, and we also applied a post-processing operation to clear peripheral voxels below t_1. For 38 images the results were similar to those for controls, and their worst 5 images had the mean and standard deviation of Dice similarity for S3, S4 and S5, respectively of 94.52% ± 0.68%, 96.84% ± 0.38%, and 96.21% ± 0.87%. The other two remaining images had larger errors and required interactive repairing by the DIFT algorithm [9].

	OCM [21]		CBM [22]		CSM	
Obj.	Mean	S.dev	Mean	S.dev	Mean	S.dev
S1	96.49	0.51	97.06	1.06	97.37	0.46
S2	96.60	0.51	97.20	1.21	97.59	0.48
S3	92.31	1.22	94.39	0.95	94.97	0.84
S4	95.47	0.44	96.66	1.36	97.16	0.48
S5	95.37	0.60	96.43	1.16	97.05	0.55

Table 1: Mean (%) and standard deviation (%) of the Dice similarity.

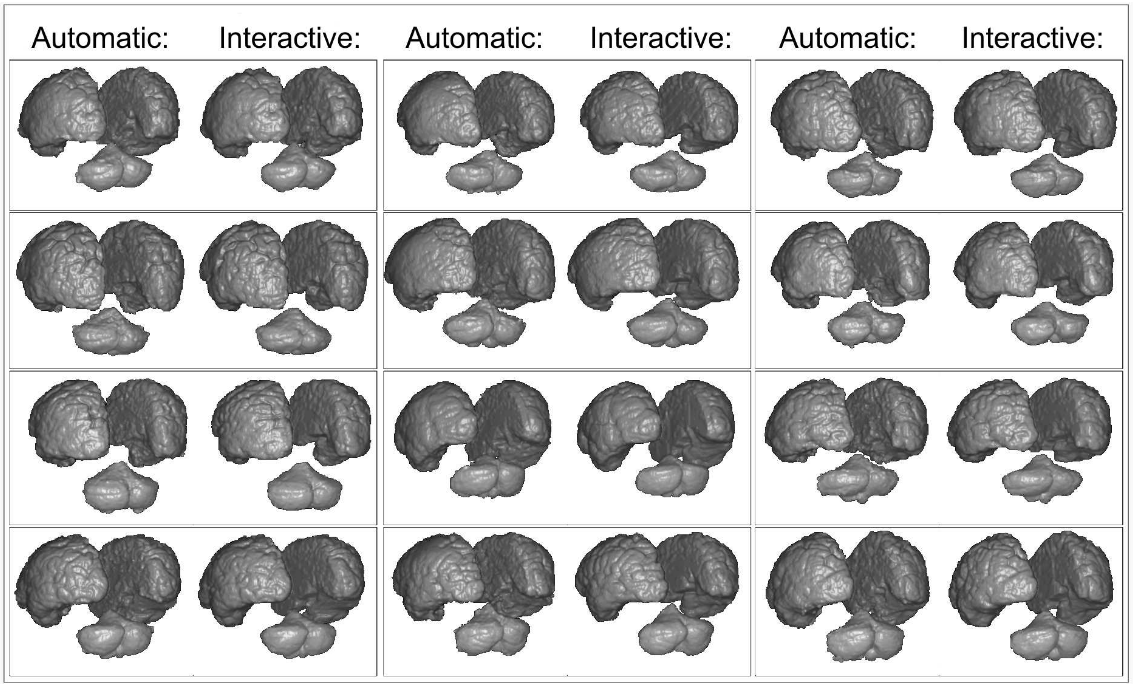

Figure 9: Segmentation results by *Cloud System Model* and the provided ground truth, shown as 3D renditions.

5 Conclusion

We have presented model-based approaches for automatic image segmentation, which employ recognition and delineation in a tightly coupled manner. The previous approaches,

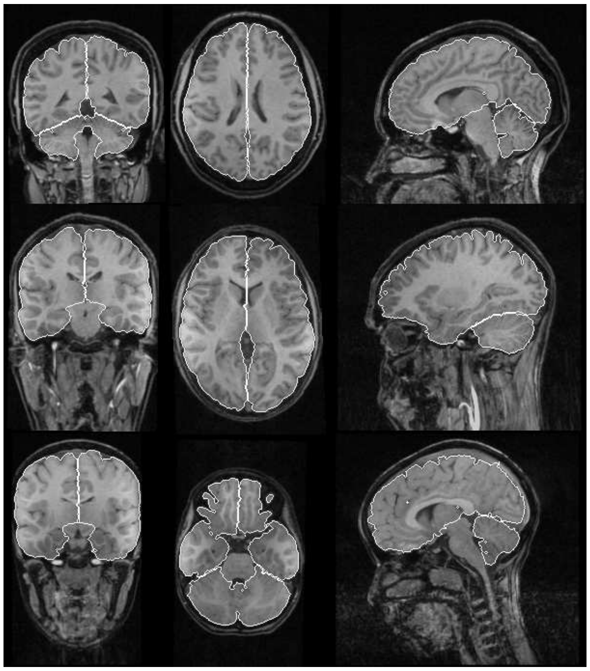

Figure 10: Sample slices from control subjects with segmentation results.

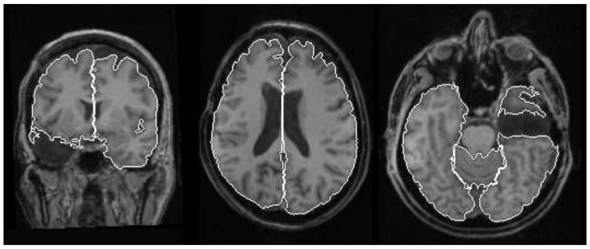

Figure 11: Sample slices from patients with segmentation results.

called OCM [21] and CBM [22], are particular cases of the proposed method, named Cloud System Model (CSM). The three methods were compared for automatic MR-T1 image segmentation of several brain structures. The results indicated that CSM is more precise and more accurate than OCM and CBM for this particular application. CSM is simpler and faster than other approaches commonly used in the literature in brain research. It can also be easily implemented in parallel, taking advantage of machines with multiple processors and cores.

Our future work will include evaluation of CSM using other medical imaging modalities, employing better quality images from a 3T MRI scanner, segmenting sub-cortical brain structures, and other image analysis applications.

Acknowledgment

The authors thank FAPESP (Proc. 05/59808-0 and Proc. 07/52015-0), and CNPq (Proc. 302617/2007-8) for the financial support, and Dr. Fernando Cendes (FCM-UNICAMP) for the images.

References

[1] F.P.G. Bergo, A.X. Falcão, P.A.V. Miranda, and L.M. Rocha. Automatic image segmentation by tree pruning. *Journal of Mathematical Imaging and Vision*, 29(2-3):141–162, 2007.

[2] F.P.G. Bergo, G.C.S. Ruppert, L.F. Pinto, and A.X. Falcão. Fast and robust mid-sagittal plane location in 3D MR images of the brain. In *Intl. Conf. BIOSIGNALS*, Funchal, Portugal, Jan 2008.

[3] Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 105–112, 2001.

[4] Xinjian Chen, Jayaram K. Udupa, Xiaofen Zheng, Drew Torigian, and Abass Alavi. Automatic anatomy recognition via multi object oriented active shape models. In *SPIE Medical Imaging*, volume 7259, Orlando, Florida, USA, Feb 2009.

[5] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models – their training and application. *CVIU*, 61(1):38–59, 1995.

[6] A.M. Dale, B. Fischl, and M.I. Sereno. Cortical surface- based analysis: I. segmentation and surface reconstruction. *NeuroImage*, 9:179–194, 1999.

[7] Nicolae Duta and Milan Sonka. Segmentation and interpretation of MR brain images using an improved knowledge-based active shape model. In *IPMI*, pages 375–380, 1997.

[8] A. X. Falcão, J. K. Udupa, and F. K. Miyazawa. An ultra-fast user-steered image segmentation paradigm: Live-wire-on-the-fly. *IEEE TMI*, 19(1):55–62, Jan 2000.

[9] A.X. Falcão and F.P.G. Bergo. Interactive volume segmentation with differential image foresting transforms. *IEEE TMI*, 23(9):1100–1108, 2004.

[10] A.X. Falcão, J. Stolfi, and R.A. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE TPAMI*, 26(1):19–29, 2004.

[11] R. S. J. Frackowiak, K. J. Friston, C. Frith, R. Dolan, C. J. Price, S. Zeki, J. Ashburner, and W. D. Penny. *Human Brain Function.* Academic Press, 2nd edition, 2003.

[12] K.A. Ganser, H. Dickhaus, R. Metzner, and C.R. Wirtz. A deformable digital brain atlas system according to Talairach and Tournoux. *Medical Image Analysis*, 8(1):3–22, Mar 2004.

[13] V. Grau, A.U.J. Mewes, M. Alcaniz, R. Kikinis, and S.K. Warfield. Improved watershed transform for medical image segmentation using prior information. *IEEE TMI*, 23(4):447–458, Apr 2004.

[14] J. Larsson. *Imaging vision: functional mapping of intermediate visual processes in man.* PhD thesis, Karolinska Institutet, 2001. ISBN 91-7349-090-3.

[15] L. Liang, K. Rehm, R.P. Woods, and D.A. Rottenberg. Automatic segmentation of left and right cerebral hemispheres from MRI brain volumes using the graph cuts algorithm. *NeuroImage*, 34(3):1160–1170, 2007.

[16] F. Lijn, M. Bruijne, Y.Y. Hoogendam, S. Klein, R. Hameeteman, M. Breteler, and W.J. Niessen. Cerebellum segmentation in MRI using atlas registration and local multi-scale image descriptors. In *Proc. of the ISBI*, Boston, MA, 2009. to appear.

[17] J. Liu and J.K. Udupa. Oriented active shape models. *IEEE TMI*, 28(4):571–584, 2009.

[18] A. Madabhushi and J.K. Udupa. Interplay between intensity standardization and inhomogeneity correction in mr image processing. *IEEE TMI*, 24(5):561–576, 2005.

[19] J. Mangin, J. Regis, and V. Frouin. Shape bottlenecks and conservative flow systems. In *IEEE Work. MMBIA*, pages 319–328, San Francisco, CA, 1996. IEEE Computer Society Press.

[20] P.A.V. Miranda and A.X. Falcão. Links between image segmentation based on optimum-path forest and minimum cut in graph. *Journal of Mathematical Imaging and Vision*, 2009. doi: 10.1007/s10851-009-0159-9.

[21] P.A.V. Miranda, A.X. Falcão, and J.K. Udupa. *CLOUDS*: A model for synergistic image segmentation. In *Proc. of the ISBI*, pages 209–212, Paris, France, May 2008.

[22] P.A.V. Miranda, A.X. Falcão, and J.K. Udupa. Cloud bank: A multiple clouds model and its use in MR brain image segmentation. In *Proc. of the ISBI*, Boston, MA, 2009. to appear.

[23] P.A.V. Miranda, A.X. Falcão, and J.K. Udupa. Synergistic arc-weight estimation for interactive image segmentation using graphs. *Computer Vision and Image Understanding*, 2009. doi: 10.1016/j.cviu.2009.08.001.

[24] W. L. Nowinski. Anatomical targeting in functional neurosurgery by the simultaneos use of multiple schaltenbrand-warhen brain atlas microseries. *Medical Image Analysis*, (71):103–116, 1998.

[25] A. Parraga, A. Susin, J. Pettersson, B. Macq, and M. De Craene. 3D atlas building in the context of head and neck radiotherapy based on dense deformation fields. In *XX SIBGRAPI*, pages 321–328, Oct 2007.

[26] Volker Stix. Finding all maximal cliques in dynamic graphs. *Comput. Optim. Appl.*, 27(2):173–186, 2004.

[27] S. Wang and J.M. Siskind. Image segmentation with minimum mean cut. In *ICCV*, volume 1, pages 517–525, 2001.