Semi-automatic 3D segmentation of brain structures from MRI

Qing He*, Kevin Karsch and Ye Duan

Department of Computer Science,
University of Missouri-Columbia,
Columbia, MO 65211, USA
E-mail: qhgb2@mizzou.edu
E-mail: krkq35@gmail.com
E-mail: duanye@missouri.edu
*Corresponding author

Abstract: We present a semi-automatic 3D segmentation method for brain structures from Magnetic Resonance Imaging (MRI). There are three main contributions. First, our method combines boundary-based and region-based approaches but differs from previous hybrid methods in that we perform them in two separate phases. This allows for more efficient segmentation. Second, a probability map is generated and used throughout the segmentation to account for the brain structures with low-intensity contrast to the background. Third, we develop a set of tools for manual adjustment after the segmentation. This is particularly important in clinical research because the reliability of the results can be ensured. The experimental results and validations on different data sets are shown.

Keywords: semi-automatic segmentation; MRI; magnetic resonance imaging; brain structures; deformable model.

Reference to this paper should be made as follows: He, Q., Karsch, K. and Duan, Y. (2011) 'Semi-automatic 3D segmentation of brain structures from MRI', *Int. J. Data Mining and Bioinformatics*, Vol. 5, No. 2, pp.158–173.

Biographical notes: Qing He is a PhD candidate in the Department of Computer Science, University of Missouri-Columbia. She obtained her ME from Shanghai Jiaotong University in 2006 and BE from Nanjing University of Science and Technology in 2004. Her current research area includes medical image segmentation, statistical shape analysis and biometric feature localisation.

Kevin Karsch received a BS in Computer Science and Mathematics from the University of Missouri in 2009. Currently, he is pursing a PhD in Computer Science at the University of Illinois at Urbana-Champaign. His research interests include computer graphics and computer vision.

Ye Duan is an Associate Professor of Computer Science at the University of Missouri-Columbia. He received his BA in Mathematics from Peking University in 1991. He received his MS in Mathematics from Utah State University in 1996. He received his MS and PhD in Computer Science from the State University of New York at Stony Brook in 1998 and 2003, respectively. From September 2003 to August 2009, he was an Assistant Professor of Computer Science at the University of Missouri-Columbia. His research interests include computer graphics and visualisation, biomedical imaging and

computer vision, geometric and physics-based modelling, virtual reality and human–computer interaction, computer animation and simulation.

1 Introduction

In recent years, medical image segmentation has become a standard technique for visualising structures of the human brain as well as performing various types of volumetric and shape comparisons among these structures. Since the introduction of medical image segmentation, many methods have been implemented for brain structure segmentation from MRI. These methods can be categorised into manual, semi-automatic and fully automatic methods. Manual segmentation is tedious, requires much attention to detail, and the results are not reproducible. On the other hand, fully automatic methods require no user interaction and are completely reproducible for the same data, but the reliability of these methods is often an issue because it solely depends on the method and no human knowledge can be of any help. Therefore, semi-automatic methods have become the preferred type of medical image segmentation (Hu et al., 2008).

Semi-automatic segmentation usually requires the user to initialise a point or region inside the object to be segmented. Region-based and boundary-based techniques are two main categories of semi-automatic segmentation. Region-based approach (Zhu et al., 1995) assigns membership to pixels/voxels according to homogeneity statistics. Since there is no easier way to distinguish boundaries and interior pixels/voxels of the object, this method can lead to noisy boundaries and holes in the interior. Boundary-based technique (Staib and Duncan, 1992; Kass et al., 1987; McInerney and Terzopoulos, 1999) attempts to align an initial deformable boundary with the object boundary by minimising an energy functional, which quantifies the gradient features near the boundary. The main drawback of these methods is their sensitivity to the initial conditions. To avoid being trapped in local minima, most of these algorithms require the model to be initialised near the solution or supervised by high-level guidance, thus initialisation plays a crucial role in the results of the segmentation. Several hybrid methods combine region-based and boundary-based approaches to overcome the disadvantages of each approach alone. Ronfard (1994) used region-based information to drive the explicit deformable models in their techniques, while Chakraborty and Duncan (1995), Jones and Metaxas (1997a, 1997b, 1998) and Chen and Metaxas (2005) have addressed these issues by interlacing region-based and boundary-based methods into a united, iterative segmentation process. The efficacy of these types of algorithms exceed that of region-based or boundary-based methods independently, but the most notable disadvantages of these methods are that they are limited to slice-by-slice (2D) segmentation and have relatively low efficiency compared with other segmentation algorithms.

This paper presents a semi-automatic method that combines region-based and boundary-based approaches. It differs from previous hybrid methods in that it implements region-based and boundary-based approaches in two separate phases, which allows more efficient segmentation and effectively avoids local minimum. After a simple user initialisation, a region-based technique is used to generate an initial seed contour that roughly represents the actual boundary, and a boundary-based method is used to guide the subsequent contour deformation. Since the seed generated by the region-based

method is already close to the target boundary, the time needed for the subsequent boundary-based method is greatly reduced. The region-based method usually takes much less time than the boundary-based method, so our method is more efficient than a boundary-based method alone.

For some brain structures such as the thalamus whose boundaries are not defined by gradient, a general boundary-based method cannot obtain the boundary accurately even with a good initial seed. Chan and Vese (2001) proposed a level-set method that can detect objects whose boundaries are not necessarily defined by gradient. Our contour deformation scheme is based on Chan and Vese (2001) with some modifications. Heckenberg et al. (2006) proposed a semi-automatic segmentation method using explicit active contour. A non-parametric kernel-based method was used to simultaneously update the interior probability distribution during the model evolution to overcome low-contrast boundaries. We incorporate this idea into part of our segmentation framework such that the user can choose to create a binary image based on Heckenberg et al. (2006) or proceed with the original image.

Segmentation accuracy is crucial in clinical research. Hence, an expert is often needed to go through every segmentation result and make necessary adjustment. The most significant disadvantage of fully automatic methods is the lack of expert control over the segmentation, while semi-automatic methods have more flexibility by virtue of user intervention. We develop a user-friendly interface for user initialisation and visualisation. Moreover, we design a set of tools, which allows the user to make modifications to the results after segmentation. In this way, an expert can make the results more reliable and practical in clinical applications by post-segmentation adjustments. The paper is organised as follows. Section 2 gives an overview of our method and the user interface. Section 3 describes our segmentation algorithms in detail. Section 4 shows the experimental results, Section 5 shows the validation of the results and Section 6 concludes the paper.

2 Overview

The overall segmentation framework is shown in Figure 1. The processes in round rectangles are operated by the user, and those in ellipses are automatic procedures. Depending on the intensity feature of the brain structure to be segmented, the user can choose to initialise a point or a small region inside the structure. If the user initialises a point, the subsequent seed initialisation and the deformation are performed on the original image. This is usually suitable for structures with high-intensity contrast to the background, such as the Corpus Callosum (CC). If the user initialises a region, a binary image is generated using the method in Heckenberg et al. (2006) and the subsequent procedures are based on the binary image, which is updated during the deformation. This is useful for structures with similar intensity as the background such as the thalamus. Region-based methods are used to generate the initial seed, which is as close to the true boundary as possible, and boundary-based methods (level set in our case) are used to refine the segmentation result by deformation. The details of the seed generation and the deformation will be discussed in the following sections. After the deformation, manual adjustment can be performed before or after the mesh extraction if the segmentation result is not perfect. The mesh extraction step converts the implicit volume representation

to a triangle mesh for the ease of later analysis. A standard marching cubes algorithm (Lorensen and Cline, 1987) is applied here.

Figure 2 shows the user interface of the segmentation software we have developed. It is designed to be simple and easy to manipulate. It displays three orthogonal views of a volumetric image, along with a 3D view. The tools for user initialisation and manual adjustment are on the right side.

Figure 1 Flow chart of the segmentation framework

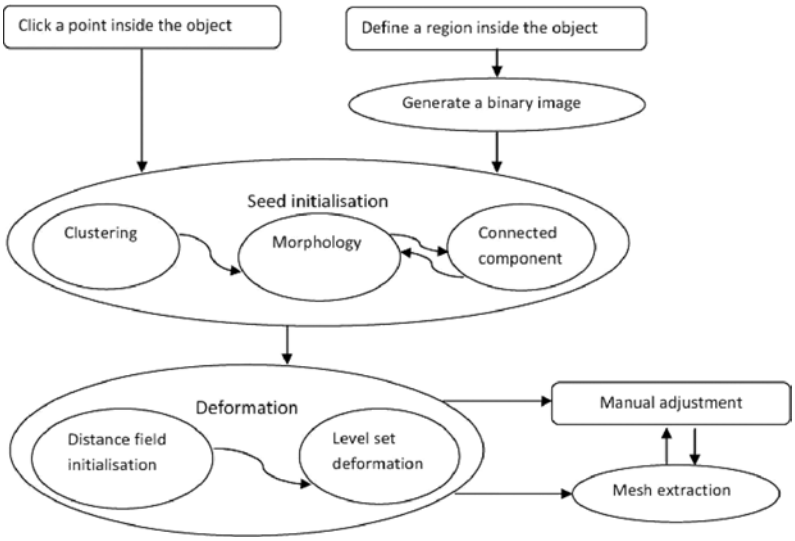
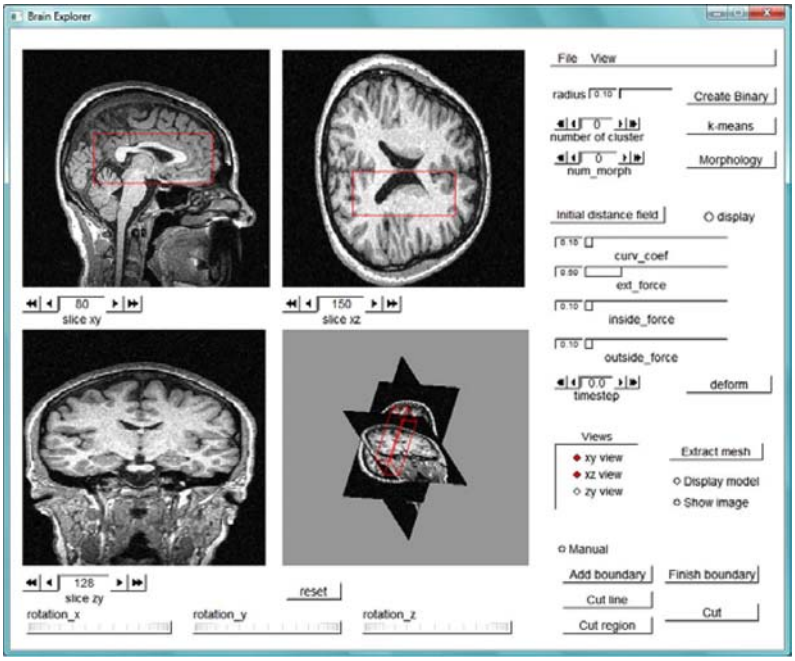


Figure 2 User interface of the segmentation software (see online version for colours)



3 Methods

In this section, we describe the details of the method. We first focus on the case when no binary image is generated, and then generalise our methods to the case of using the binary image. Finally, we describe the user interface for manual adjustment after the segmentation.

3.1 Seed generation

To improve the efficiency of the computation, the user can define a Region of Interest (ROI) before initialising the seed. An easy way to do this is to draw a rectangle on two of the orthogonal views and the two rectangles define a 3D ROI. Figure 2 shows the ROI of the ventricle defined on sagittal and coronal views. All the subsequent operations are performed within the ROI. If the ROI is not defined by the user, the entire image is the default ROI.

We use a region-based method to generate an initial seed, which is close to the object boundary. The user is required to click a point $\vec{x}_0$ inside the target structure on one of the three orthogonal views. After that, clustering is performed based on the image intensity. We adopt fast adaptive *k*-means (Darken and Moody, 1990) with a modified distance measure. Instead of L2 square distance, Delta-MSE dissimilarity (Xu, 2004) is used as the distance measure. The Delta-MSE dissimilarity between data point $\vec{x}_i$ and the cluster centroid $\vec{c}_j$ is defined as

$$D(\vec{x}_i, \vec{c}_j) = w_{ij} \left\| \vec{x}_i - \vec{c}_j \right\|^2, \quad w_{ij} = \begin{cases} n_j /(n_j + 1), & p(i) \neq j \\ n_j /(n_j - 1), & p(i) \neq j \end{cases} \tag{1}$$

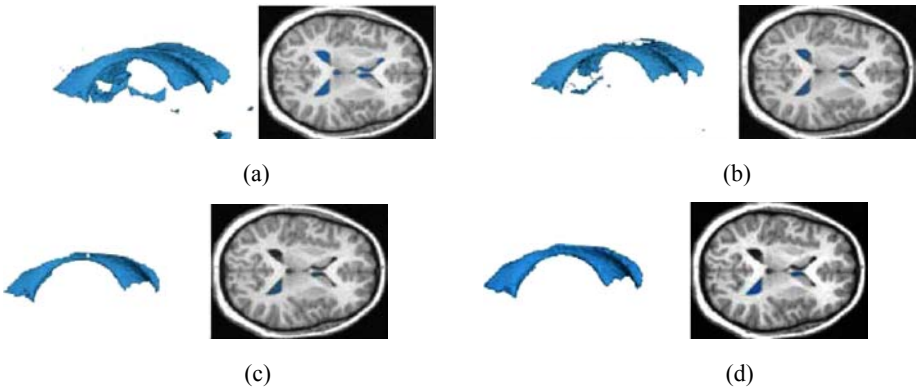
where $p(i)$ is the class label of $\vec{x}_i$. The weight makes the dissimilarity bigger than L2 square distance if $\vec{x}_i$ is allocated in the cluster and smaller otherwise. Therefore, the reassignments of data samples into their closest clusters are driven with Delta-MSE dissimilarity more frequently than with L2 square distance, enabling a faster convergence to the global optimum.

The cluster that contains $\vec{x}_0$ is the most important cluster because it provides an initial estimate of the target shape. The number of clusters (k) is a parameter that the user can twist. However, in most cases, $k = 3$ or $k = 4$ can guarantee a good estimate of the target shape. Let Ω_i. denote the voxels contained in the cluster that $\vec{x}_0$ belongs to and Ω_o denote the voxels outside the cluster. The next steps will further refine the seed (Ω_i) by ensuring its connectedness and removing voxels from the seed, which are not strongly affiliated with the target structure. Voxels that are not affiliated with the target structure will in most cases be in or near the boundary of Ω_i, denoted as $\partial\,\Omega_i$. To remove these voxels, Ω_i is eroded m times with a mathematical morphology operation. This essentially removes $\partial\,\Omega_i$ from Ω_i at each iteration, and then recalculates Ω_i after each step of erosion. In some cases, essential voxels of the target structure are removed from Ω_i, but the deformation stage (see Section 3.2) is used to overcome these seeding artefacts.

As with most mathematical morphology operations, Ω_i should then be dilated the same number of steps as it was eroded. However, we insert a connected components step before dilation to ensure the connectedness of Ω_i. It is possible that there

are some other regions in the ROI that have similar intensity to the target structure but are not connected to it, so they are more likely to be in the same cluster with the target structure. These artefacts can be removed by applying the connected components algorithm (Ronsen and Denijver, 1984). The algorithm is implemented originating from $\vec{x}_0$ so that the connected component containing $\vec{x}_0$ can be found. During the connected components search, only voxels in Ω_i are available to be searched. Ω_i is recalculated based on all voxels that are visited during the search, and the outside set is also recalculated accordingly. Finally, the set Ω_i is dilated m times as it was eroded, recalculating Ω_i and Ω_o dynamically. The value of m depends on the specific structure of interest, which will be discussed in our experimental results. These seed creation steps can be done very efficiently, and allow for the development of a seed that is roughly equivalent to the target structure. Figure 3 shows the seed initialisation phase of a right lateral ventricle from start to finish.

Figure 3 Demonstration of the seed generation phase of a right lateral ventricle. The 3D seed (left) is rendered explicitly for display purpose. On the 2D image slice (right), pixels are rendered as blue if they belong to the seed: (a) seed after K-means clustering; (b) seed after erosion; (c) seed after connected components search; (d) seed after dilation (see online version for colours)



3.2 Level-set deformation

The seed created in the previous section is coarse and contains various artefacts or imperfections due to image noise. This is the main drawback of region-based methods in image segmentation. Therefore, the seed needs to be deformed under a boundary-based mechanism to increase the accuracy of the segmentation. However, since this seed is already close to the target boundary, local maxima can be avoided in the following deformation, and the time needed for the deformation is greatly reduced compared with a general point or sphere seed.

The seed Ω_i is transformed into an implicit function and deformed based on a level-set PDE. A signed distance function ϕ is initialised such that $\phi(\vec{x})$ is the signed Euclidean distance (negative on Ω_i and positive on Ω_o) from $\vec{x}$ to the closest voxel in $\partial\,\Omega_i$. This is implemented by a fast sweeping method (Zhao, 2005). ϕ is initially very large at all voxels. The values for voxels in or adjacent to $\partial\,\Omega_i$ are directly computed with the following equation:

$$\phi(\vec{x}) = \begin{cases} \min_{\vec{y}\in\partial\Omega_i} \| \vec{x} - \vec{y} \|, & \vec{x} \in \Omega_o \\ -\min_{\vec{y}\in\partial\Omega_i} \| \vec{x} - \vec{y} \|, & \vec{x} \in \Omega_i \end{cases}. \tag{2}$$

The values of the rest voxels are computed by propagating an estimation of the actual distance using previous computed elements. The details of the propagating algorithm can be found in Zhao (2005).

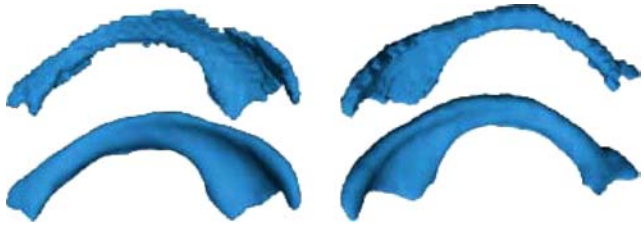
ϕ is then deformed based on a PDE similar to those described in Chan and Vese (2001) Gibou and Fedkiw (2005). The PDEs in these works focus heavily on information other than edges to drive the level-set deformation. The idea behind the deformation is to introduce an artificial time variable t and to update the level-set function ϕ as time elapses. We update ϕ by numerically solving the following PDE:

$$\partial\phi/\partial t = \alpha\nabla\left(\nabla\phi/\left\|\nabla\phi\right\|\right) - \beta - \gamma_1(I - \bar{\Omega}_i)^2 + \gamma_2(I - \bar{\Omega}_o)^2 \tag{3}$$

where $\bar{\Omega}_i$ and $\bar{\Omega}_o$ are the average intensities of the voxels in the sets Ω_i and Ω_o, respectively. α, β, γ_1 and γ_2 are weight parameters. Specifically, α controls the curvature term, β is an external force to be applied to ϕ, γ_1 and γ_2 are the weights for the distance functions $(I - \bar{\Omega}_i)^2$ and $(I - \bar{\Omega}_o)^2$, respectively. The sets Ω_i and Ω_o are updated at each iteration to be consistent with their definitions, i.e., $\phi(\vec{x}) < 0$ for $\vec{x} \in \Omega_i$ and $\phi(\vec{x}) > 0$ for $\vec{x} \in \Omega_o$.

For the purpose of efficiency, we also implement a narrow band algorithm (Lefohn et al., 2004) so that only values of the distance field that are within a certain threshold are updated. That is, we only solve equation (2) near the voxels where $\phi(\vec{x}) = 0$ when updating the distance field. The deformation will stop when an equilibrium is achieved. Figure 4 shows the comparison of the results before and after deformation.

Figure 4 Seed contours rendered before (top) and after (bottom) level-set deformation: (a) right lateral ventricle and (b) left lateral ventricle (see online version for colours)



(a) (b)

3.3 Binary image generation

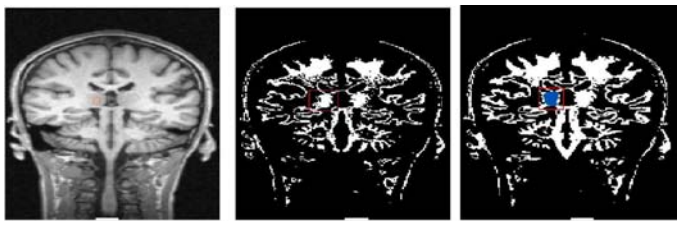
In the case that a binary image is desired, the user can define a sphere centred at $\vec{x}_o$ with the radius r (the circle on the 2D view in Figure 5(a)). The initial binary image is generated using a non-parametric method (Heckenberg et al., 2006). Briefly, the intensity distribution is estimated based on Parzen-window function with Gaussian kernel. Suppose the volume of the initial sphere region is V, the probability of a voxel's intensity value i being consistent with the interior intensity inside the seed can be derived as

$$P(i \mid M) = \frac{1}{V} \iiint \frac{1}{\sqrt{2\pi\sigma}} e^{-(i-I(y))^2/2\sigma^2} \tag{4}$$

where σ is a constant that specifies the width of the Gaussian kernel, y is a voxel and the integration is over the image domain. A threshold (usually the mean probability over the entire image domain) is applied to produce a binary image, where voxels with probability higher than the threshold have value 1 and other voxels have value 0 (Figure 5(b)).

The following seed generation procedure is the same as in Section 3.1 except that it is performed on the binary image, so the seed before the deformation is generated based on the initial binary image in Figure 5(b). In this case, we can always let $k = 2$ for the k-means clustering because there are only two intensity levels in the binary image. During the deformation, the binary image is updated at each iteration based on the most recent distance field (Figure 5(c)). This method is especially suitable for brain structures with low contrast to the background. The difference between our method and the method in Heckenberg et al. (2006) is that we apply region-based techniques on the binary image before the deformation, so that the seed is close to the target boundary, while in Heckenberg et al. (2006) a point seed is deformed from the beginning.

Figure 5 (a) Initial seed of the thalamus defined by the user (circle on the 2D slice); (b) initial binary image generated from the seed in (a); (c) updated binary image during the deformation with the level-set model overlaid (blue region) (see online version for colours)



(a) (b) (c)

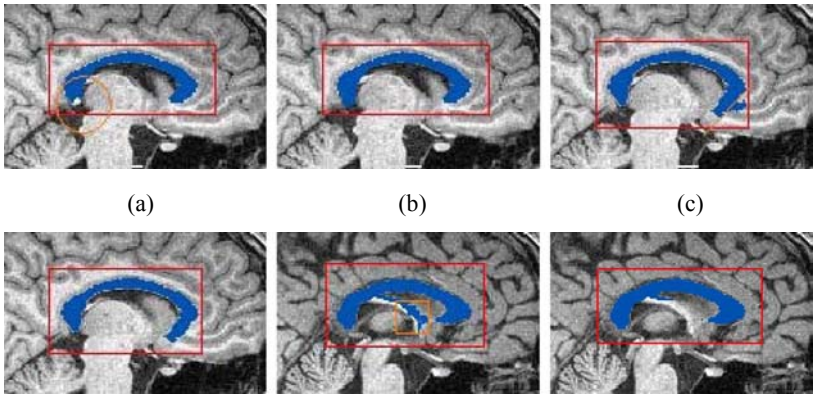
3.4 Manual adjustment

In some extreme cases, the above-mentioned level-set model cannot achieve a perfect segmentation, and manual adjustment is necessary to ensure the segmentation correctness. We design several convenient tools for manual adjustment after the automatic segmentation.

In the case that part of the brain structure is missing in the segmentation result (Figure 6(a)), the user can delineate the boundary of the missing part by sequential point clicking. The user-delineated boundary and the partial boundary of the original segmentation result form a closed region, and the voxels inside this region are moved to set Ω_i from set Ω_o. This is done by setting $\phi(\vec{x}) < 0$ for these voxels (Figure 6(b)).

If the result includes voxels outside of the structure of interest, two options can be performed. The user can draw a cutting line on a 2D slice (Figure 6(c)) and specify which side of the line is to be cut off. The voxels on this side of the line are moved to set Ω_o if they were in set Ω_i before (Figure 6(d)). The other option is to draw a rectangle that encloses the voxels that need to be cut off from the segmentation result (Figure 6(e) and (f)).

Figure 6 Demonstration of manual operations on the segmentation of the corpus callosum: (a) the splenium bottom is missing (the circle is used to highlight the missing part); (b) the result after manual modification on (a); (c) the segmentation result extends the anterior boundary and the user-specified cut line; (d) the result after cutting off the region on right side of the line in (c); (e) the fornix enclosed in the small rectangle is to be cut off and (f) the result after cutting off the fornix in (e) (see online version for colours)



(a) (b) (c)

(d) (e) (f)

All the manual operations have to be performed on a slice-by-slice basis, which might be time-consuming. However, the purpose of manual adjustment is to ensure the correctness of the segmentation result, and it is easier to make judgements on each 2D slice instead of a 3D model. Moreover, manual operation is not the main part of our segmentation framework and there is limited amount of operation needed, so the time spent on manual adjustment is not a big issue. The experimental results will show the frequency of the data and the slices that need manual adjustment.

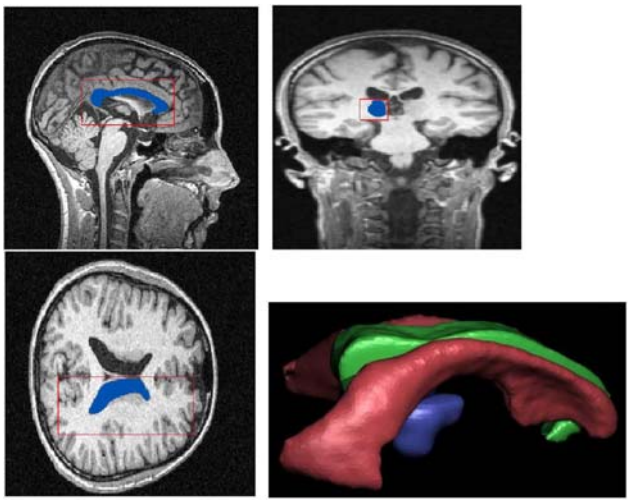
4 Results

The segmentation algorithm is implemented in C++ and the user interface is developed using FLTK (http://www.fltk.org/). We test our method on two data sets. The first data set consists of brain MRI scans of 20 college student volunteers obtained on a 3T Siemens Trio scanner with a standard 8-channel head coil in the Department of Psychological Sciences at the University of Missouri (TR = 1920 ms, TE = 3.75 ms, flip angle = 8°, in-plane resolution = 1 × 1 mm, slice thickness = 1 mm, number of images = 160, matrix = 256 × 256). The second data set consists of 25 children with autism recruited from the Thompson Center for Autism and Neurodevelopmental Disorders. T1-weighted brain MR images were acquired using the Siemens Symphony 1.5 T scanner (TR = 35 ms, flip-angle = 30 degrees, thickness = 1.5 mm, matrix = 512 × 512).

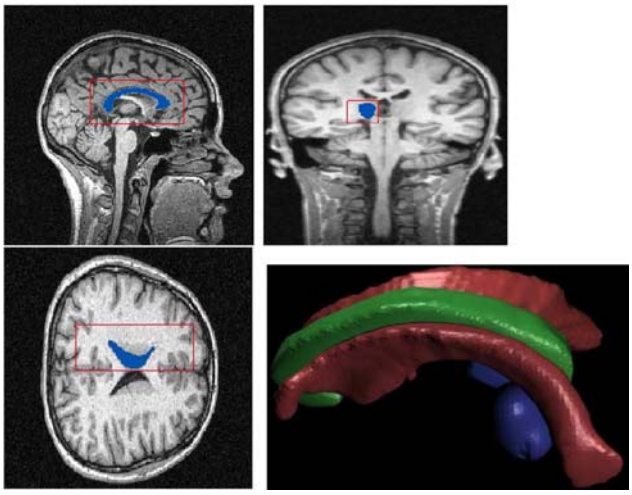
Three structures are segmented using our method: the CC, the lateral ventricles and the thalami. Figure 7 shows the segmentation results (after manual adjustment if necessary) in 2D and 3D views of 3 randomly selected subjects from each data set. For each structure, one orthogonal view that best displays the features of this structure is shown, i.e., sagittal view for the CC, coronal view for the ventricles and axial view for

the thalami. The segmentation result is overlaid on the corresponding orthogonal view. A 3D view of all segmented structures is also shown. For the CC and the ventricles, we choose point initialisation described in Section 3.1, and for the thalami we choose region initialisation described in Section 3.3. The deformation parameters are set in the following range: $\alpha = 0.02\sim0.06$, $\beta = 0.02\sim0.05$, $\gamma_1 = \gamma_2 = 0.1$, time-step = 0.2. The seed generation parameters depend on the intensity distribution of the segmented structure and the background. We find that the best parameter values for the CC are: $k = 3$, $m = 1$, for the ventricles: $k = 3$, $m = 0$, and for the thalami: $r = 1.9\sim2.2$, $k = 2$, $m = 1$.

Figure 7 Segmentation results of student volunteers (a)–(c) and autistic children (d)–(f). Top left: sagittal view of the segmentation result of the CC (blue region); top right: axial view of the segmentation result of the thalamus; bottom left: coronal view of the segmentation result of the ventricle; bottom right: 3D view of the segmented structures (see online version for colours)

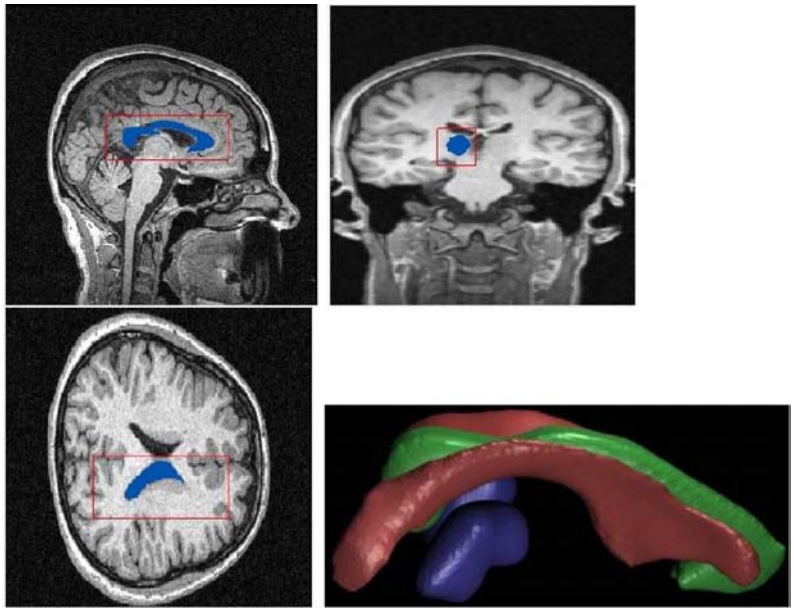


(a)

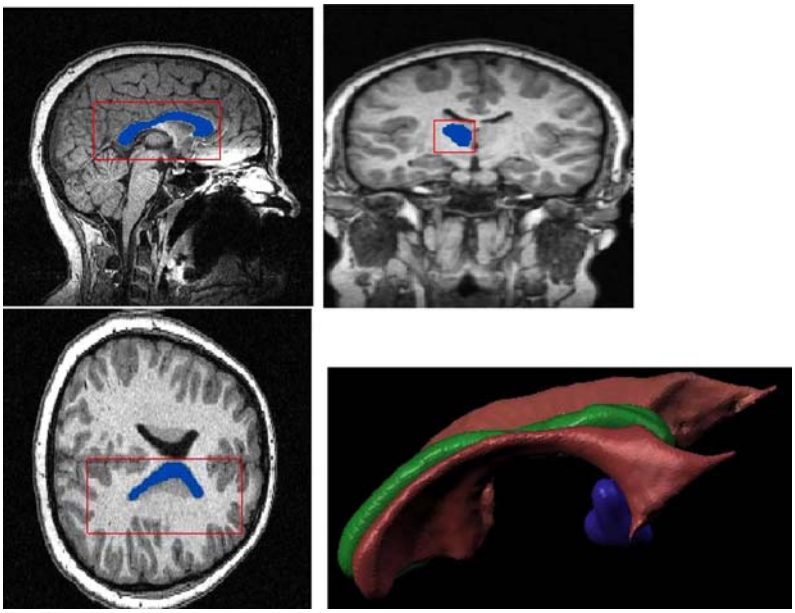


(b)

Figure 7 Segmentation results of student volunteers (a)–(c) and autistic children (d)–(f). Top left: sagittal view of the segmentation result of the CC (blue region); top right: axial view of the segmentation result of the thalamus; bottom left: coronal view of the segmentation result of the ventricle; bottom right: 3D view of the segmented structures (see online version for colours) (continued)

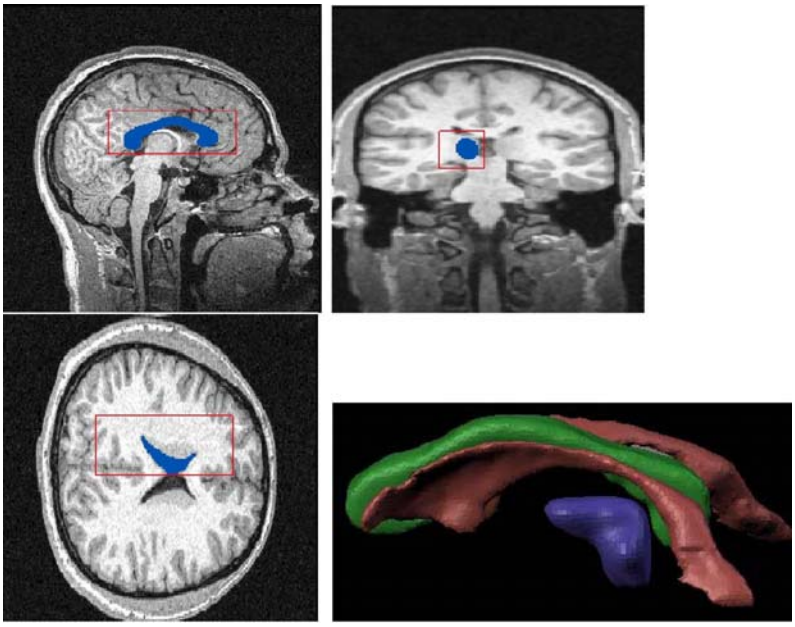


(c)

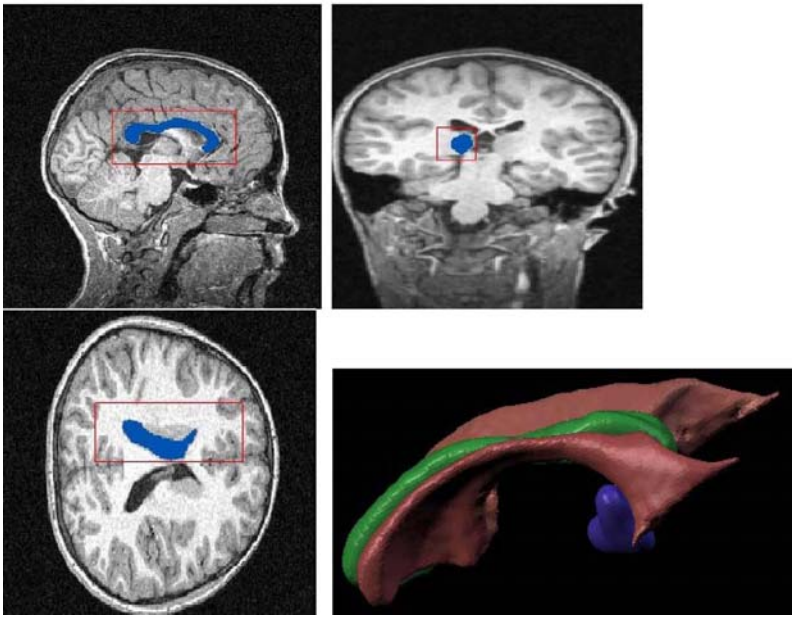


(d)

Figure 7 Segmentation results of student volunteers (a)–(c) and autistic children (d)–(f). Top left: sagittal view of the segmentation result of the CC (blue region); top right: axial view of the segmentation result of the thalamus; bottom left: coronal view of the segmentation result of the ventricle; bottom right: 3D view of the segmented structures (see online version for colours) (continued)



(e)



(f)

With the facility of manual adjustment, our segmentation software can achieve both efficiency and optimal reliability, which is desired in clinical applications. Manual adjustment is made by a knowledgeable expert using our manual tools in Section 3.4 if necessary. From our 45 data sets, no manual adjustment is needed for the ventricle segmentation, 10 data sets need manual adjustment for the CC segmentation and 12 need manual adjustment for the thalamus segmentation. The slices that need to be modified usually appear at the end of the structure in its dominant view (e.g., sagittal view for the CC). Among the data that need manual adjustment, the average number of slices adjusted is 6 for the CC and 4 for the thalamus.

5 Validation

To test the accuracy and efficiency of the segmentations, we compare our method with a generic boundary-based method (which uses a sphere seed centred at the user-specified point). The only difference between these two methods is the way that the seed is created. The segmentation results of the two methods are both compared with manual (ground truth) segmentations. The manual segmentation is performed by a trained expert different from the one who does the manual adjustment. For the comparisons, we calculate Dice similarity and overlap coefficients between manual segmentations and the results of each method. The range of these statistics lies between 0 and 1, with 1 indicating a perfect agreement between our segmentation and the manual segmentation. Table 1 provides a summary of the results. Overall, both methods are able to closely achieve the accuracy of the manual segmentations, but our method has higher accuracy than the generic boundary-based method. Manual segmentation takes 2 h for each individual structure on average. To segment one structure using our method, the user initialisation takes 20 s, the automatic algorithm part takes 90 s and the manual adjustment takes 10 min on average. Therefore, our segmentation framework is much more efficient than manual segmentation, even when the manual adjustment is needed. We also show the efficiency of our segmentation method by comparing the running time of the automatic segmentation part (after user initialisation and before manual adjustment) between the two methods. The results are shown in Table 2, which indicates that our method is much faster in the automatic part than the generic method. Given the previous comparison of the accuracy, it is demonstrated that our new seeding scheme increases both accuracy and efficiency.

Table 1 Accuracy measurements of our results and the results from the generic boundary-based method

	Dice				*Overlap*			
	Our method		*Generic method*		*Our method*		*Generic method*	
Structure	*Mean*	*Std.*	*Mean*	*Std.*	*Mean*	*Std.*	*Mean*	*Std.*
CC	0.79	0.05	0.76	0.05	0.79	0.03	0.75	0.03
Left ventricle	0.81	0.04	0.78	0.05	0.73	0.05	0.72	0.05
Right ventricle	0.78	0.06	0.75	0.06	0.81	0.03	0.78	0.03
Left thalamus	0.69	0.08	0.64	0.09	0.66	0.06	0.63	0.07
Right thalamus	0.70	0.07	0.67	0.09	0.65	0.05	0.64	0.07

Table 2 Automatic segmentation time of our method and the generic boundary-based method (in seconds)

Structure	*Our method*	*Generic boundary-based method*
CC	98	240
Left ventricle	75	182
Right ventricle	78	186
Left thalamus	88	124
Right thalamus	84	120

Since our method is not fully automatic, human factors may affect the consistency of the segmentation results. To test the effect of human factors, we have two operators to perform the segmentation and manual adjustment on the same subset of the data. Twenty data sets are randomly selected as the subset of the data. The two operators have been trained enough to be able to perform manual adjustment properly. Each operator is blind to the seed initialisation and ROI definition of the other. The overlap of each pair of the segmentation results is measured by Dice coefficient. Table 3 shows the statistics of the Dice coefficients on different segmented structures. The high Dice similarity shows that our method can produce consistent results despite the user initialisation.

Table 3 Dice coefficients of the segmentation results from two raters

Structure	*Mean*	*Std.*
CC	0.89	0.02
Left ventricle	0.94	0.01
Right ventricle	0.90	0.01
Left thalamus	0.82	0.07
Right thalamus	0.85	0.08

6 Conclusion and future work

We present a semiautomatic 3D segmentation framework for brain MRI, which combines region-based and boundary-based approaches. After user initialisation, a region-based technique is used to generate an initial seed contour that roughly represents the actual boundary, and a boundary-based method is used to guide the subsequent contour deformation. To deal with brain structures with low-intensity contrast to the background, we choose to generate a binary image iteratively based on the intensity distribution of the deformable model, thus facilitating the segmentation. A user-friendly interface is developed for user initialisation and visualisation. Moreover, a set of tools are designed that allows the user to make modifications to the results after segmentation. The results show that our method has similar accuracy to the manual segmentation but is much faster even with manual adjustment after the segmentation.

Our segmentation framework is a practical tool in clinical research. Our method can overcome the unreliability of fully automated methods in that we allow manual

adjustment to the results, and it is much faster than pure manual segmentation. The user initialisation is also very simple for a general user.

Currently, the segmentation software is limited to segment one structure at a time. However, it is straightforward to extend it to segment multiple structures at the same time. Although our primary interest is the segmentation of brain MRI, our segmentation framework can also be used for other imaging modalities and anatomical regions, such as lung CT images. To further put it into practice, we aim at developing a web application of our segmentation framework in the future.

Acknowledgements

This work is supported in part by an NIH pre-doctoral training grant for Clinical Biodetectives, a Thompson Center Research Scholar fund, the Department of Defense Autism Concept Award and the NARSAD Foundation Young Investigator Award.

References

Chakraborty, A. and Duncan, J.S. (1995) 'Integration of boundary finding and region-based segmentation using game theory', in Bizais, Y. *et al.* (Eds.): *Information Processing in Medical Imaging*, Kluwer, pp.189–201.

Chan, T.A. and Vese, L.A. (2001) 'Active contour without edges', *IEEE Transactions of Image Processing*, Vol. 10, No. 2, pp.266–277.

Chen, T. and Metaxas, D. (2005) 'A hybrid framework for 3D medical image segmentation', *Medical Image Analysis*, Vol. 9, pp.547–565.

Darken, C.J. and Moody, J. (1990) 'Fast adaptive kmeans clustering: some empirical results', *Proceedings International Conference on Neural Networks*, IEEE Computer Science Press, New York, Vol. 2, pp.233–238.

Gibou, F. and Fedkiw, R. (2005) 'A fast hybrid k-means level set algorithm for segmentation', *4th Annual Hawaii International Conference on Statistics and Mathematics*, Honolulu, Hawaii, pp.281–291.

Heckenberg, G., Xi, Y., Duan, Y. and Hua, J. (2006) 'Brain structure segmentation from MRI by geometric surface flow', *International Journal of Biomedical Imaging*, Vol. 2006, Article ID 86747, pp.1–6.

Hu, Y.J., Grossberg, M.D. and Mageras, G.S. (2008) 'Semiautomatic medical image segmentation with adaptive local statistics in conditional random fields framework', *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Vancouver, BC, pp.3099–3102.

Jones, T. and Metaxas, D. (1997b) 'Automated 3D segmentation using deformable models and fuzzy affinity', *Proceedings of the 15th International Conference on Information Processing in Medical Imaging*, Poultney, Vermont, USA, pp.113–126.

Jones, T.N. and Metaxas, D.N. (1997a) 'Segmentation using deformable models with affinity-based localization', *Proceedings of the First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Computer-Assisted Surgery*, Grenoble, France, pp.53–62.

Jones, T.N. and Metaxas, D.N. (1998) 'Image segmentation based on the integration of pixel affinity and deformable models', *Proc. Conf. Computer Vision and Pattern Recognition*, Santa Barbara, CA, USA, pp.330–337.

Kass, M., Witkin, A. and Terzopoulos, D. (1987) 'Snakes: active contour models', *Intl. Journal of Computer Vision*, Vol. 1, pp.321–331.

Lefohn, A.E., Kniss, J.M., Hansen, C.D. and Whitaker, R.T. (2004) 'A streaming narrow-band algorithm: interactive computation and visualization of level sets', *IEEE Transactions on Visualization and Computer Graphics*, Vol. 10, No. 4, pp.422–433.

Lorensen, W.E. and Cline, H.E. (1987) 'Marching cubes: a high resolution 3D surface construction algorithm', *ACM SIGGRAPH Computer Graphics*, Vol. 21, No. 4, pp.163–169.

McInerney, T. and Terzopoulos, D. (1999) 'Topology adaptive deformable surfaces for medical image volume segmentation', *IEEE Transactions on Medical Imaging*, Vol. 18, No. 10, pp.840–850.

Ronfard, R. (1994) 'Region based strategies for active contour models', *International Journal of Computer Vision*, Vol. 13, No. 2, pp.229–251.

Ronsen, C. and Denijver, P.A. (1984) *Connected Components in Binary Images; The Detection Problem*, John Wiley & Sons, Inc., New York, NY.

Staib, H.L. and Duncan, J.S. (1992) 'Boundary finding with parametrically deformable models', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 11, pp.1061–1075.

Xu, M. (2004) 'Delta-MSE dissimilarity in GLA-based vector quantization', *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'04)*, Montreal, Canada, pp.V-813–V-816.

Zhao, H.K. (2005) 'A fast sweeping method for eikonal equations', *Mathematics of Computation*, Vol. 74, pp.603–627.

Zhu, C.S., Lee, T.S. and Yuille, A.L. (1995) 'Region competition: unifying snakes, region growing, and Bayes/mdl for multiband image segmentation', *Proc. Intl. Conf. on Computer Vision*, Boston, MA, USA, pp.416–423.