Efficient Technique for Corpus Callosum Segmentation in Midsagittal Brain MR Images

by

Yue Li

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Signal and Image Processing

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Neurological disorders are among major causes of disability in Canada. In the diagnostic procedure, Magnetic Resonance Imaging (MRI) is commonly used as it is non-invasive and can produce dramatic contrast between different brain tissues. Brain tissue segmentation is a fundamental step in brain MR images analysis.

Corpus Callosum (CC) is an important brain tissue and is always adopted as the landmark of human brain. In this thesis, we propose an intelligent computer-aided detection (CAD) system for automatic segmentation of CC in T1-weighted midsagittal brain MRI slices. The proposed CAD system has three modules: *Adaptive Mean Shift Clustering (AMS)*, *Automated CC Contour Initialization (ACI)*, and *Geometric Active Contour (GAC) based Segmentation*. In the first module, homogenous regions in the input image are divided into clusters with an adaptive mean shift clustering method. In the second module, area analysis, template matching, shape and location analysis are used to identify the cluster that contains CC and extract a rough boundary of

CC as the initial contour. In the last module, the boundary of recognized CC region is used as the initial contour in the GAC model, and is evolved to obtain the final segmentation result of CC. Experimental results demonstrate that the proposed AMS-ACI technique is able to provide accurate initial CC contour, and the proposed AMS-ACI-GAC technique overcomes the problem of user-guided initialization in existing GAC techniques, and provides a reliable and accurate performance in CC segmentation.

Acknowledgement

Foremost, I would like to express my deep gratitude to my supervisor, Prof. Mrinal Mandal, who has given me a valuable opportunity to carry out medical image processing research, and provided me continuous support, unlimited encouragement and patience in my M.Sc. study and research. With his immense knowledge and guidance, I have gained huge helps in all of the time in my research and thesis writing. I cannot imagine that I could have a better supervisor and mentor for my M.Sc. study and research, a better role model for my entire life.

I would like to thank my co-supervisor, Prof. S. Nizam Ahmed, for the clinical support in my research work, and for his help in teaching me about anatomy and morphology of human brain, interpretation of brain Magnetic Resonance (MR) images, knowledge about epilepsy and its typical Magnetic Resonance Imaging (MRI) findings, and in collecting brain MRI data.

I am grateful for Dr. Nasir Rizvi for his help in preparing ground truth of CC on brain MRI data. I thank my fellow labmates in Multimedia

Computing and Communications Laboratory (MCCL): Gencheng Guo, Tao Xu, Cheng Lu, Hongming Xu, Huiquan Wang, Salah Al-Heejawi, Bingjie Ma, Ying Wang, Xiaohui Gong and Lu Chang, for inspiring discussions and helpful advice in my research and study, and for warm help in my life.

I would like to thank Dr. Roger Zemp and Dr. Jie Chen for being my defense committee and devoting their time in reviewing my thesis and providing their insightful comments, valuable advice and enlightening questions for my research.

Last but not the least, I would like to thank my family: my parents Dan Chen and Bingpei Li, my grandparents, Yiping Zhang and Guoqing Li, for their endless love and encouragement throughout my life, and my daughter Joycelyn, who always brings me hope and love during my difficulties.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Acronyms	Definition
ACI	Automatic CC Contour Initialization
ACM	Active Contour Model
AMS	Adaptive Mean Shift
BMRI	Brain MRI
CAD	Computer-Aided Detection or Computer-Aided Diagnosis
CC	Corpus Callosum
CT	Computed Tomography
EEG	Electroencephalogram
GAC	Geometric Active Contour
GUI	Graphic User Interface
LB	Laplace-Beltrami
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
NCC	Normalized Cross Correlation
OASIS	Open Access Series of Imaging Studies

PDF	Probability Density Function
SVM	Support Vector Machine
WHO	World Health Organization

List of Symbols

Symbols	Definition
I	A grayscale brain MR image
p_i	The i^{th} pixel in image I
V_i	The grayscale intensity of p_i
N	The total number of pixels in the brain MR image I
V_{cur}	The initial feature point in the AMS iteration
K	The number of neighbors that is considered in calculating the adaptive bandwidth for V_{cur}
$V_{cur,K}$	The K^{th}-nearest neighbor of V_{cur} in the feature space
$\|E\|_1$	L1 norm of expression E
$\|E\|$	Absolute value of expression E
h_{cur}	The adaptive bandwidth for V_{cur} in the AMS clustering
S_{cur}	A symmetric window with bandwidth h_{cur} centered at V_{cur}
J_{cur}	The number of feature points included in S_{cur}
$M_h(V_{cur})$	The weighted mean shift vector within window S_{cur}
d	The dimension of feature space

V_j	The j^{th} point within window S_{cur}
G	The uniform kernel function
c	The normalization constant that makes the integral of $G(x)$ equals to one
T_{AMS}	A predefined threshold used as a stopping criterion for search of a mode in the AMS clustering
p	The proportion of K in N
S_{CC}	The specific cluster that includes the CC tissue
C_i	The i^{th} cluster obtained by the AMS clustering
Cl_{Cand}	The candidate CC clusters satisfying the area criterion
P_C	The number of pixels included in a cluster
T_{Thr}	The predefined threshold of the number of pixels included in the CC clusters
I_B	The generated binary image after applying the AMS clustering and area analysis
A	The largest connected area (i.e., the area with largest number of pixels) in a region
C	The initial contour of CC area
$p_{in}(z,\phi)$	The *pdf* of interior intensities of the contour C
$p_{out}(z,\phi)$	The *pdf* of exterior intensities of the contour C

ε	The expected value of a functional
TP	The number of pixels in the true positive area (image region which is correctly classified as CC)
TN	The number of pixels in the true negative area (image region which is correctly classified as background)
FP	The number of pixels in the false positive area (image region which is incorrectly classified as CC)
FN	The number of pixels in the false negative area (image region which is incorrectly classified as background)

Chapter 1

Introduction

Computer-Aided Detection or Computer-Aided Diagnosis (CAD) systems are procedures that may help clinicians interpret various medical images. Typically, the CAD system combines artificial intelligence, digital image processing techniques and related knowledge of disease diagnosis. In diagnostic radiology, medical image modalities such as X-ray, Ultrasound, MRI and histopathological images complement clinical diagnosis. With the help of CAD system, computer-aided image analysis results based on computer algorithms could potentially increase the sensitivity of identifying abnormal lesions and structures.

The CAD systems are currently employed in the investigation of tuberculosis [1] and breast cancer [2]. The model used to segment CC can be applied to segment areas of subtle abnormalities in the brain, thus increasing diagnostic sensitivity and accuracy.

1.1 Status of Epilepsy in the World and in Canada

Epilepsy is one of the most common chronic brain disorder affecting almost 50 million people around the world. Epilepsy is the tendency to have recurrent unprovoked seizures. The clinical presentation is dependent on the underlying brain structure that is involved. Epilepsy can affect both social and mental well-being, and patients with epilepsy have two or three times higher risk of premature death compared to the general population. According to a recent report from Epilepsy Canada, about 15,500 people are diagnosed with epilepsy every year in Canada [3]. It is therefore important to study and understand epilepsy with a multidisciplinary approach. MRI images help to identify underlying structural abnormalities as a cause of seizure. CAD techniques therefore can facilitate to increase the sensitivity of such diagnosis.

1.2 Diagnosis of Epilepsy

The diagnosis of epilepsy is predominantly clinical based on the clinical history provided by the patient, family members and other witnesses. Electroencephalograms (EEGs) are useful to classify the seizures and the MRI helps to identify underlying structural abnormalities. If a clear epileptogenic focus is identified, the patient may be investigated to determine his/her candidacy for epilepsy surgery.

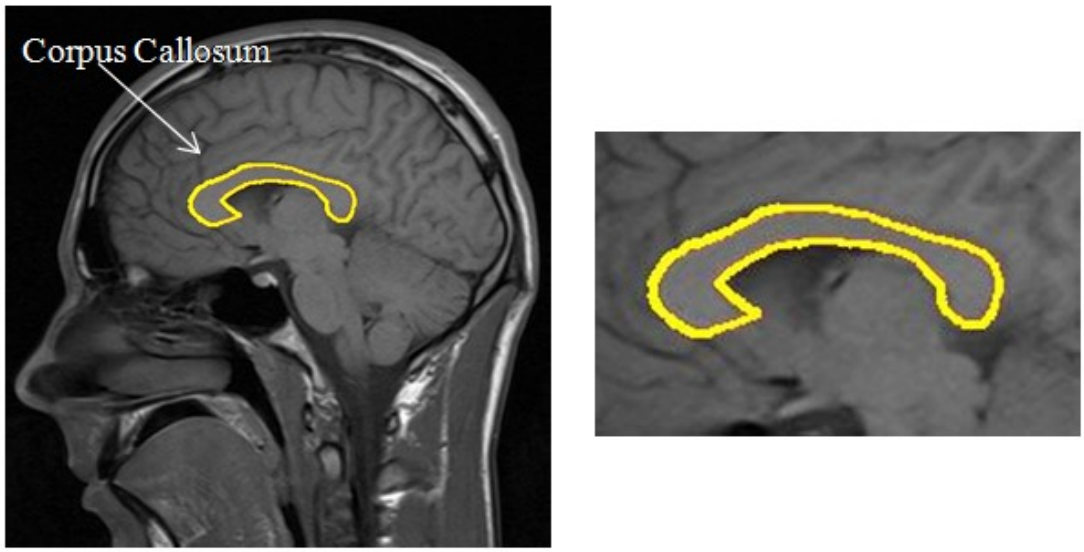
MRI is an imaging technique that is widely used to obtain images of the brain structures. It uses powerful magnets to align the nuclei of atoms, primarily hydrogen,

in the body [4]. When additional energy (in the form of radio wave) is added to the magnetic field, the magnetic vector is deflected and the hydrogen nuclei resonate. The nuclei produce rotating magnetic files of their own. Such rotating magnetic fields can be detected by a scanner and used to generate an image. The brain MRI has no radiation exposure to the subjects, and can produce dramatic contrast between different brain structures. Compared with other neuroimaging techniques, brain MRI can produce clearer and more detailed images of the brain.

The CC is a large white-matter structure in the human brain that connects the left and right cerebral hemispheres, and transfers sensory, motor and cognitive information between the two hemispheres.

Corpus Callosum can be easily identified on visual inspection. We therefore decided to apply our CAD technique to identify this structure as a proof of principle and the landmark in human brain.

Typically, the CC is examined from a midsagittal MRI slice, where the CC region could be best viewed. An example of the T1-weighted midsagittal brain MRI slice is illustrated in Fig. 1.1 (a), and a close-up example of CC region is illustrated in Fig. 1.1 (b). Note that the yellow contours in both figures indicate the CC region. It is observed that the CC region is located near the centre of the brain, and always presents homogeneous high intensity, and has the appearance of broad arched band. The location and shape features of CC region are important features that can be used to distinguish CC from other brain tissues.



(a) **(b)**

Fig. 1.1. Corpus Callsoum in a T1-weighted midsagittal brain MR image. (a) The CC is highlighted by yellow contour. (b) A close up example of the CC, and the thick yellow contour indicates its area.

1.3 Problem Statement and Motivation

Traditionally, the neuroimages are examined by professional radiologists, and the brain tissue and its change are manually separated from surrounding brain tissues based on their observation, knowledge and experience. Major difficulties in segmentation of brain tissues in the subcortical regions arise from low contrast and filed inhomogeneity. Note that the accuracy for detection of epileptogenic regions using the neuroimaging techniques may be influenced by radiologist's subjective bias. Because of fatigue, overlook or overloaded images, epileptogenic regions with subtle structural changes might be missed. In addition, the inter-reader and intra-reader variations often occur [5]. In order to address these problems, it is of great value to develop an automated CAD system to perform automatic CC segmentation.

The intention of our work is to segment CC automatically on both patients' MR images and normal MR images. The automatic classification of patients' images and normal images using the segmented CC is not discussed in our work. Major difficulties in segmentation of brain tissues in the subcortical regions, however, arise from low contrast and field inhomogeneity.

The objective of this thesis was to develop an efficient CAD system for automated detection of CC based on T1-weighted midsagittal MR images from patients with epilepsy, which will provide the ground work for utilizing a similar technique to identify abnormal brain structures with the potential of neurological disorders.

1.4 Contribution and Thesis Organization

This thesis focuses on the automatic detection and segmentation of CC in the T1-weighted midsagittal brain MRI slice. The major contributions are listed as follows:

- An automatic technique for CC segmentation in T1-weighted midsagittal brain MR images has been proposed (presented in Chapter 3).
- A graphical user interface of the proposed CC segmentation technique has been developed (presented in Chapter 4).

The rest of the thesis is organized as follows. Chapter 2 presents a review of related techniques. Chapter 3 introduces the details of proposed CC segmentation technique. Chapter 4 discusses the development of the graphical user interface of the proposed technique, as well as the segmentation performance evaluation. Chapter 5 presents the conclusion and the potential future research directions.

Chapter 2

Background and Related Works

In this chapter, we briefly introduce the structure of CC. We also present a review of the state-of-the-art techniques related to the brain MRI segmentation.

2.1 Structure of CC

The CC is the largest white matter structure in human brain, around 200 to 250 million neural fibers are included in the CC [6]. It is the brain structure that connects the left and right cerebral hemispheres. [7]. The neural fibers of the CC connect the neural fibers of both the left and the right cerebral hemispheres, and the interhemispheric communications are processed through the CC [8]. The CC can be viewed as the network connecting two strong computer processors (left and right cerebral hemispheres).

The CC is wide and flat neural fiber bundle, with the appearance of broad arched band [9]. It is located near the bottom of the longitudinal fissure in human brain [10]. The structure of CC can be divided into four portions: genu, splenium,

body and rostrum. To be specific, the genu is the anterior portion of the CC, the splenium is the posterior portion of the CC, the body is the portion between the genu and the splenium, and the rostrum is the portion of the CC that extends posteriorly and inferiorly from the anterior most genu [11]. An illustration of four portions of CC structure on the midsagittal image of the brain is displayed in Fig. 2.1.

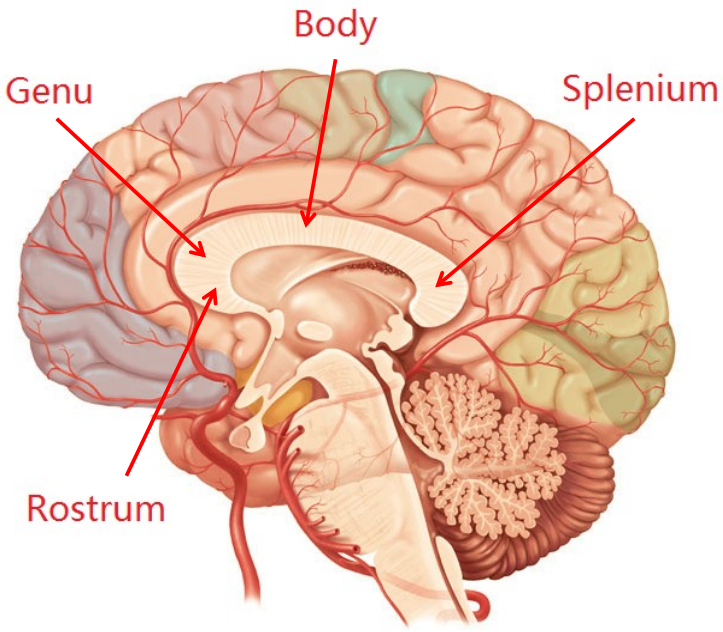


Fig. 2.1. Illustration of the genu, splenium, body and rostrum on the midsagittal image of the brain [12]

2.2 Review of Segmentation Techniques

Major difficulties in segmentation of brain tissues in the subcortical regions, however, arise from low contrast and field inhomogeneity. Several works have already been conducted in this area.

Most of the techniques are based on Active Contour Model (ACM). Brejl *et al.* [13] proposed an automated model based technique for segmentation of CC in

sagittal brain MRIs. The technique first constructs shape model and border appearance model of the objects from training images. Based on these two statistical models representing the object of interest, a shape-variant Hough transform technique is applied to approximately localize the object, and then a cost function can be automatically designed and used in the segmentation criteria of edge-based segmentation methods. This technique does not need initial close-to-target localization. However, it is likely to fail if a large variability exhibits in shape and edge appearance of the objects. Ginneken *et al.* [14] proposed an active shape model-based technique for the segmentation of CC. This technique is steered by optimal local features, chosen by statistical analysis of training images from a set of local image structure descriptors. After that, a non-linear K-nearest neighbours classifier and selected features are used to calculate the optimal displacements of landmarks. This technique has the limitation of model matching. It cannot detect patterns which are not included in the given example images. Jacob *et al.* [15] proposed an ACM based technique for CC segmentation. This technique introduces a robust gradient energy term which represents the gradient direction and has the advantage of being parameter independent, as well as a new internal energy term which forces the active contour to the constant arc-length parameterization. Experiments of segmenting CC on MRIs show good segmentation results. However, the initialization of contour is required.

Sandhu *et al.* [16] proposed a Geometric Active Contour (GAC) technique for CC segmentation. The technique incorporates the image intensity probability density functions of the background and object into active contour framework, and

evolution is steered by seeking the maximum distance of above two distributions. Although promising segmentation results on brain MRIs has been achieved, this technique is highly sensitive to initial contour. Zhou *et al.* [17] proposed an automated segmentation technique for rodent brain tissues in MRIs. This technique applies support vector machines (SVMs) to obtain prior shape knowledge of objects of interest, and incorporates an automatic shape selection into existing active shape model framework for Cerebellum, Neocortex, CC, External Capsule, Caudate Putamen, Hippocampus and Ventricles in MRIs. They evaluated their method on the database with 5000 training images and 3250 test images, and obtained an average successful rate of 92.2% in classification of test images. The promising segmentation results of active shape model were shown at the end of their paper. However, low accuracies exist in CC and external capsule segmentation. El-Zehiry *et al.* [18] proposed a novel technique to segment CC from white matter on the midsagittal plane. The technique first extracted white matter using a hierarchical model combining active contour propagation and graph cut optimization to ensure that the global energy of the contour was minimal. The connected component analysis was then used to segment the CC from white matter. Finally, it investigated the difference between ratio of CC region on the midsagittal plane and the entire intracranial volume between 12 right handed dyslexic patients and 12 controls. Experimental results showed a large variability between these two groups. However, the quantitative performance evaluation of CC segmentation is not mentioned.

Lai *et al.* [19] proposed an automated method to extract the boundary of CC on brain MRIs. Firstly, they calculated the 1st nontrivial Laplace-Beltrami (LB)

eigenfuction on the white matter surface, and then used its zero level set curve to achieve the initial guess of the CC curve. Secondly, they deformed the initial curve to get the final curve with the geodesic curvature flow on the white matter surface. They validated their method based on a dataset for multiple sclerosis study with 32 images. However, they did not compare their segmentation results with ground truth.

2.3 Summary

In this chapter, a brief introduction of the structure of CC has been proposed. A review of the brain tissue segmentation techniques has also been presented.

Chapter 3

Automatic Corpus Callosum Segmentation

In this chapter, I present an automatic Corpus Callosum segmentation technique for the T1-weighted midsagittal brain MR image. The goal of the technique is to provide an accurate segmentation of the CC region for brain magnetic resonance (MR) images.

3.1 Introduction

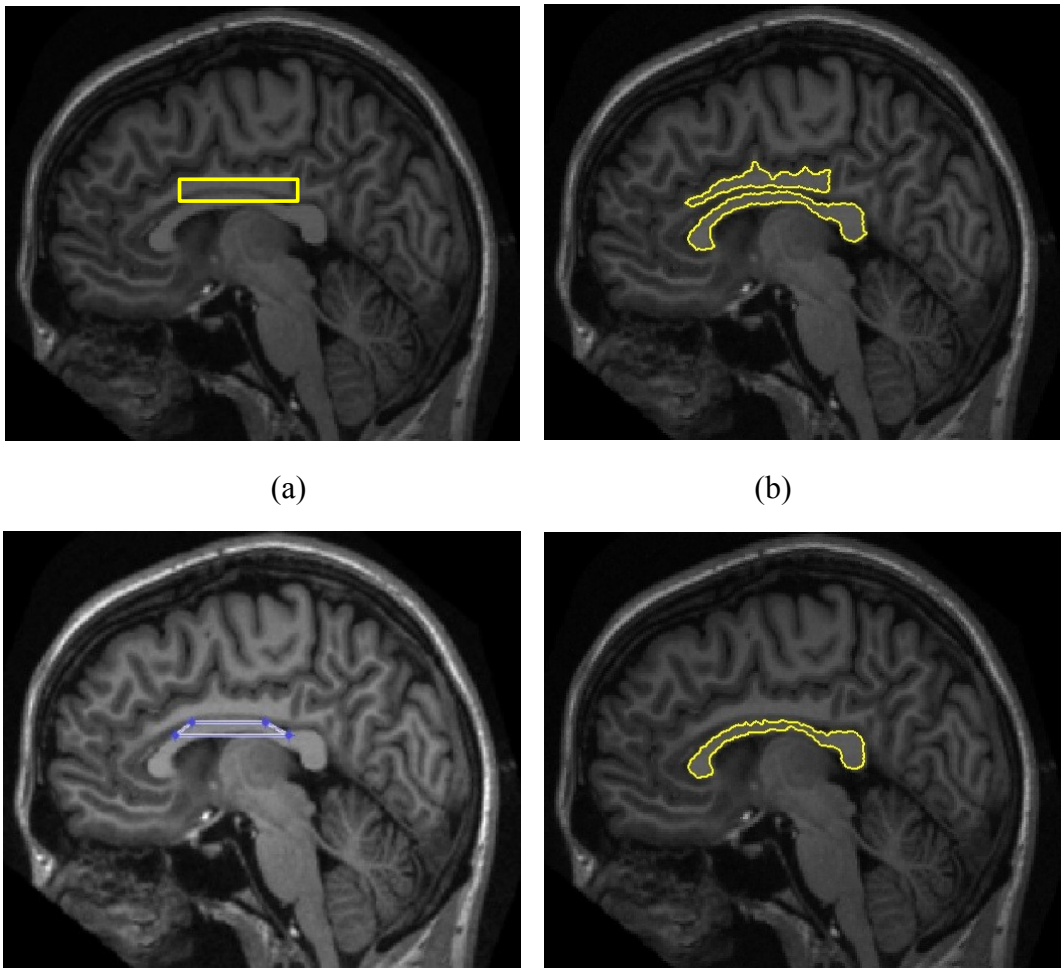
As mentioned in Chapter 1, the changes in size and shape of Corpus Callosum (CC) occur in epilepsy. Therefore, an automatic and reliable technique for segmentation of CC will facilitate the extraction of size and shape features of CC, and these features can be further used in diagnosis of epilepsy. As introduced in Chapter 2, the GAC based techniques [16] have been reported to provide promising performance of CC segmentation. However, there are still some limitations. First, initial contour is required as user inputs. Second, the GAC technique [16] is highly sensitive to initial

contour, it may fail if the initial contour is not placed appropriately. Examples of object segmentation results using different initial contours based on a same image are shown in Fig. 3.1. Fig. 3.1 (b) and (d) are the segmentation results based on initial contours shown on Fig. 3.1 (a) and (c) respectively. The yellow rectangles on Fig. 3.1 (a) and (c) indicate initial contours, and the yellow contour on Fig. 3.1 (b) and (d) indicate segmentation results.

To address the limitations of GAC based segmentation technique [16], in this chapter, we propose a fully automatic segmentation technique of CC in the midsagittal slice of T1-weighted brain MR image. The proposed technique is based on a hybrid automatic CC contour initialization (ACI) technique and the GAC technique [16]. The major contribution of the proposed technique is to provide an accurate initialization of the CC boundary. Experimental results demonstrate that the proposed technique overcomes the problem of user-guided initialization in the existing GAC technique [16], and provides a reliable segmentation performance.

3.2 The Proposed Technique

The schematic of the proposed technique is shown in Fig. 3.2. It is observed that the proposed technique contains three modules: *Adaptive Mean Shift Clustering*, *Automated CC Contour Initialization*, and *Geometric Active Contour based Segmentation.* In the first module, a set of clusters are generated from an input image based on grayscale intensity closeness using the adaptive mean shift clustering.



(a) (b)

(c) (d)

Fig. 3.1. Examples of segmentation results using different initial contour in GAC based technique [16]

In the second module, area analysis, template matching, shape and location analysis are used to identify the specific cluster that contains CC, and extract a rough boundary of CC as the initial contour. In the last module, a final CC boundary is calculated based on Active Contour Segmentation method. The details of these modules are presented in the following sections.

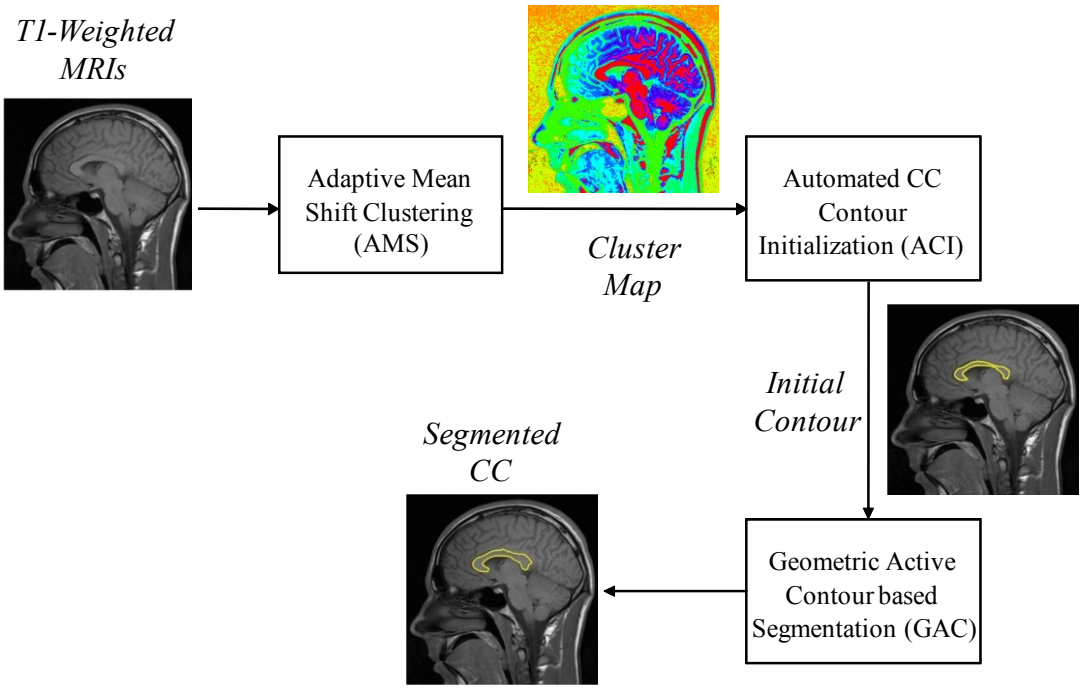


Fig. 3.2. Schematic of the proposed automatic CC segmentation technique.

3.2.1 Adaptive Mean Shift Clustering (AMS)

Based on observation in T1-weighted brain MRI (see Fig. 1.1 (a)), the appearance of CC is generally homogenous (i.e., similar grayscale intensity). For accurate segmentation, it is helpful to apply an image clustering technique as the preprocessing step. In this work, the AMS technique [20] is applied with gray-level intensities features for image clustering.

The AMS technique [20] is an iterative technique that can be used for finding modes of the probability density function (*pdf*) of the image intensity, and associating their neighboring grayscale intensities to the corresponding modes, thereby establishing clusters. Grayscale intensities associated with the same mode belong to the same cluster, and the number of clusters is the same as the number of

modes. Prior knowledge of the clusters, e.g., the number of clusters, and the shape of clusters, are not required in the AMS technique [20]. In the *pdf* of grayscale intensities of a brain MR image, there would be dense regions if this image has homogeneous areas. In other words, homogeneous areas in the image correspond to various segments in the *pdf* of grayscale intensities. For example, if there is a large homogeneous area in a grayscale image, with an intensity of about 100 (in a 8-bit scale), it could lead to a segment in *pdf* of grayscale intensities, around 100. The peak intensity of 100 is considered as a mode of the *pdf* of grayscale intensities of this image.

Note that in AMS clustering, we only consider the grayscale intensity information of the image, and do not consider the spatial information. This is because the grayscale intensity is good enough for the AMS technique to generate a rough shape of CC region. In other words, the generated clusters would be similar in intensity, and it is possible that a cluster may include several unconnected regions in the image space.

There are seven major steps of AMS clustering, which are explained in the following.

Step 1: For a brain MR image I, the *pdf* of the grayscale intensities is computed. Let p_i denote the i^{th} pixel in the image I, and V_i denote the grayscale intensity of p_i, $1 \le i \le N$, where N is the total number of pixels in the image I. Note that V_i is a mapping point of p_i in the grayscale intensity feature space of the image.

Example of an image and its corresponding *pdf* are shown in Fig. 3.3. A T1-weighted brain MRI is shown in Fig. 3.3 (a), and its *pdf* of grayscale intensities is

shown in Fig. 3.3 (b), and its magnified *pdf* of grayscale intensities in the range of [50, 120] are shown in Fig. 3.3 (c). It is observed in Fig. 3.3 (b) that several local peaks are present in the *pdf* of the grayscale intensities. Some local peaks could be better observed in Fig. 3.3 (c).

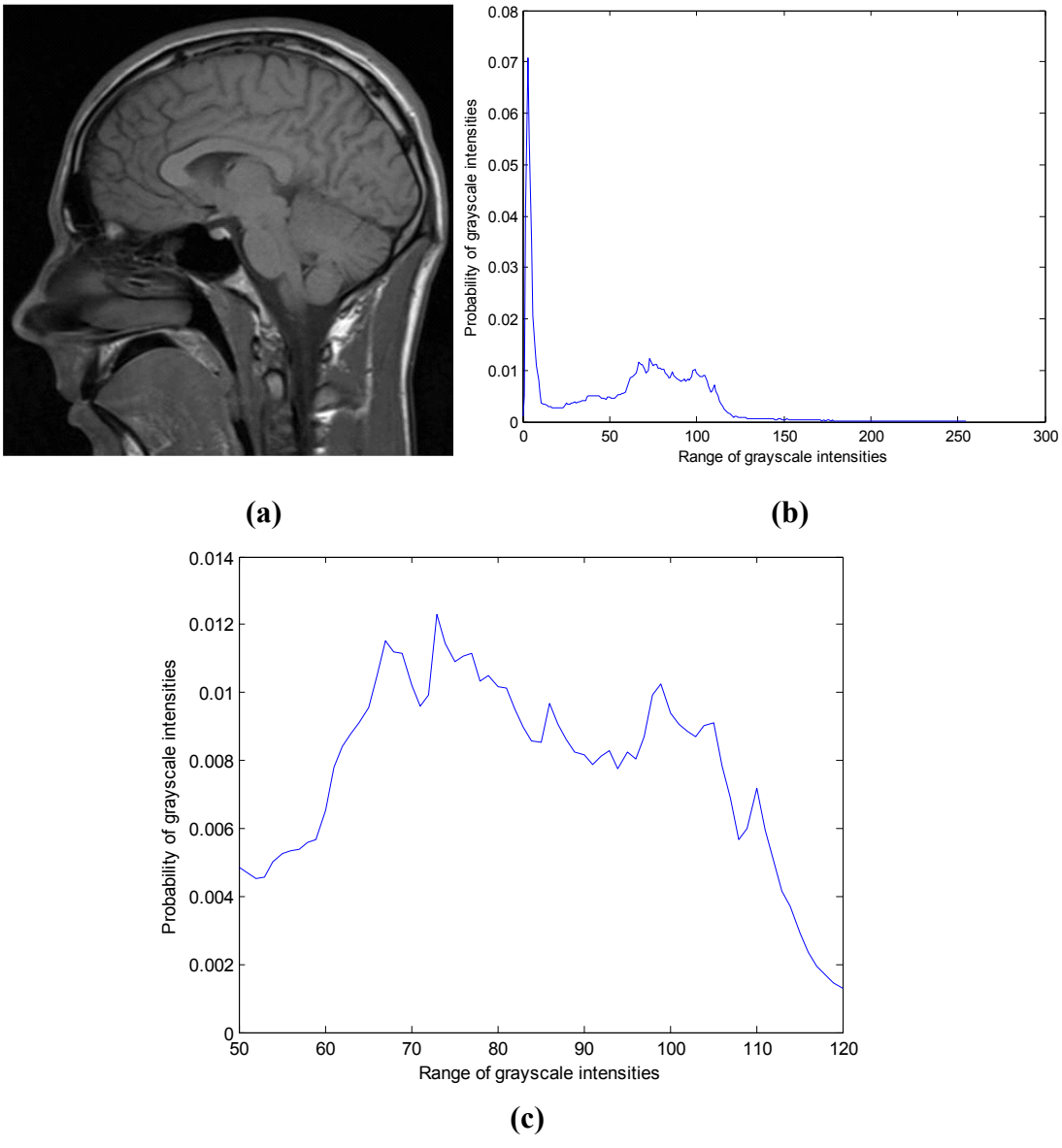


Fig. 3.3. Example of a feature space. (a) A 512×512 **brain MRI. (b)** ***Pdf*** **of the gray intensity value of the image shown in (a). (c) A blown-up version of the** ***pdf*** **shown in (b), in the range of [50 120].**

In other words, homogeneous areas with pixels that have similar grayscale intensities are present in the brain MRI. Our goal is to delineate clusters in the grayscale feature space, with the intention of clustering homogenous areas in the brain MRI. Denote the number of clusters in the image by m. Initially, m is zero. To start with, a feature point V_{cur} is initialized randomly from the feature space, and then go to step 2.

Step 2: For the point V_{cur} (a gray intensity value), the $L1$ distance (i.e., Manhattan distance) between V_{cur} and its neighbors in the feature space is calculated, and the neighbors of V_{cur} are sorted by order of increasing distance magnitude to V_{cur}. Let $V_{cur,K}$ denote the K^{th}-nearest neighbor of V_{cur}. The adaptive bandwidth h_{cur} for V_{cur} (for a given K value, K=26, 214) is calculated as follows:

$$h_{cur} = \left\| V_{cur} - V_{cur,K} \right\|_1 = \left| V_{cur} - V_{cur,K} \right| \tag{3.1}$$

where K is the number of neighbors that is considered in calculating the adaptive bandwidth h_{cur} for V_{cur}.

Step 3: For the point V_{cur}, a symmetric window S_{cur} with bandwidth h_{cur} is generated. Let the number of points included in S_{cur} be denoted by J_{cur} (including V_{cur}).

Step 4: The weighted mean shift vector $M_h(V_{cur})$ within the window S_{cur} is the distance between the current feature point V_{cur} and the center of mass within the window. $M_h(V_{cur})$ is calculated using the following equation:

$$M_h(V_{cur}) = \frac{\sum_{j=1}^{J_{cur}} \frac{1}{h_{cur}^{d+2}} (V_j - V_{cur}) G\left(\frac{V_j - V_{cur}}{h_{cur}}\right)}{\sum_{j=1}^{J_{cur}} \frac{1}{h_{cur}^{d+2}} G\left(\frac{V_j - V_{cur}}{h_{cur}}\right)} \tag{3.2}$$

where d is the dimension of feature space (we have used d=1 in this work), V_j is the j^{th} point within window S_{cur}, J_{cur} is the number of points included in the window, h_{cur} is the bandwidth for the current feature point V_{cur}. $G(x)$ is the uniform kernel function, which is defined as follows:

$$G(x) = \begin{cases} c & |x| \le 1 \\ 0 & otherwise \end{cases} \tag{3.3}$$

where c is a normalization constant such that $\int_{-\infty}^{+\infty} G(x)dx$=1.

Step 5: Determine if the following equation is satisfied,

$$|M_h(V_{cur})| < T_{AMS} \tag{3.4}$$

where T_{AMS} is a predefined threshold used as a stopping criterion for search of a mode in the current neighborhood in feature space. If Eq. (3.4) is satisfied, the current V_{cur} is stored as the converged feature point, and then algorithm goes to step 6. Otherwise, the feature point V_{cur} is updated as follows:

$$V_{cur} = V_{cur} + M_h(V_{cur}) \tag{3.5}$$

and the algorithm goes to step 2.

Step 6: The current converged point V_{cur} is stored as a mode, and a new cluster by including V_{cur} and all the points converging to V_{cur} (i.e., all the points visited by

AMS steps 2-5) is generated. Let the generated cluster be denoted by C_m (where m is the cluster number).

Step 7: In this step, we first determine if there is any pixel in the image which falls outside all the clusters that have been generated. Let us denote the set of image pixels which are not among clusters $\{C_1,\ldots,C_m\}$ by C_{rem}. If C_{rem} is a null set, the clustering stops. Otherwise, the feature point V_{cur} is initialized as follows:

$$V_{cur} = g \tag{3.6}$$

where g is a grayscale intensity, $g \in C_{rem}$, and the algorithm goes to step 2.

An example of histogram of pixels at the modes is shown in Fig. 3.4.

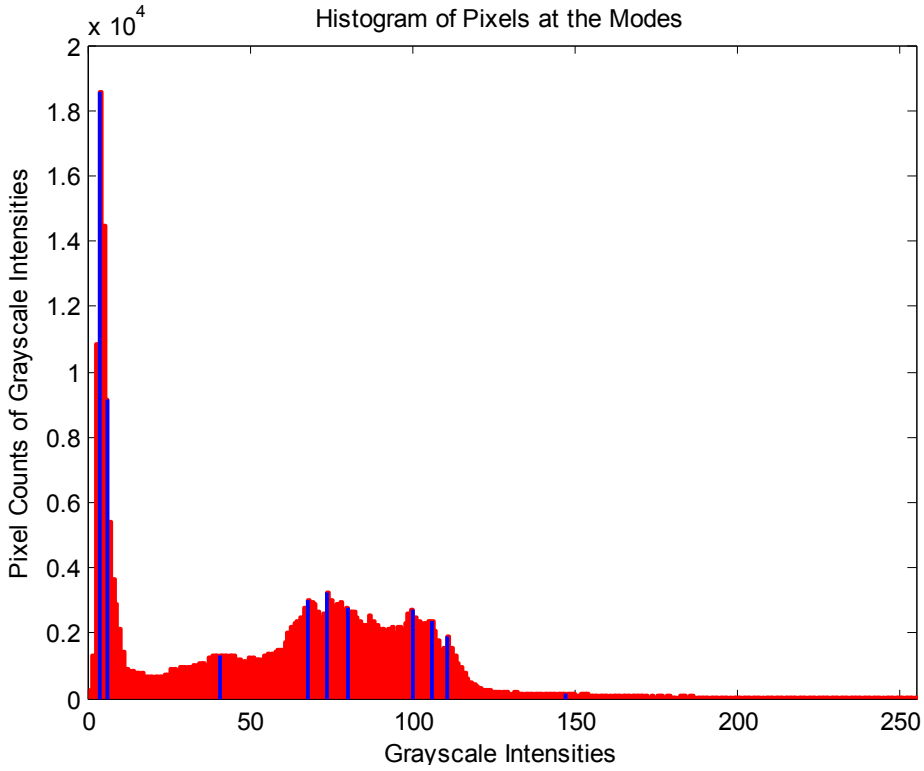


Fig. 3.4. Histogram of modes in AMS clustering obtained for the image shown in Fig. 3.3 (a). There are 10 modes with values [4, 6 41, 68, 74, 80, 100, 106, 111, 147]. The modes at 147 is not clearly visible, as there are only 117 pixels.

In this example, there are 10 modes, and each mode corresponds to a cluster. The gray intensity of a mode corresponds to representative intensity of a cluster pixel values. Examples of clusters generated in the 1st AMS iteration and in the 10th AMS iteration are shown as binary images in Fig. 3.5 (a) and Fig. 3.5 (b) respectively. In these binary images, white pixels indicate cluster points (pixels).

(a) **(b)**

Fig. 3.5. Examples of clusters (white pixels) generated in AMS iterations. (a) Cluster generated in the 1st AMS iteration. The last bandwidth $h_i = 1$, and the cluster pixel value range is [1 5]. (b) Cluster generated in the 10th AMS iteration. The last bandwidth $h_i = 108$, and the cluster pixel value range is [122, 256].

As we defined previously (see Eq. 3.1), K is the number of neighbors that is considered in calculating the adaptive bandwidth for a feature point. We calculate K using the following equation:

$$K = p \cdot N \tag{3.7}$$

where p is a constant, $0 \le p \le 1$.

An illustrative example of AMS clustering is shown in Fig. 3.6. In Fig. 3.6, the cluster maps and histograms corresponding to the image in Fig. 3.3 (a) (with $N = 262{,}144$) are shown. Figs. 3.6 (a), (c) and (e) show the cluster maps generated by the AMS, with $p = 0.08$, $p = 0.10$ and $p = 0.12$, respectively. The cluster histograms corresponding to Figs. 3.6 (a), (c) and (e) are shown in Figs. 3.6 (b), (d) and (f), respectively. It is observed that with $p = 0.08$, 0.10 and 0.12, AMS generates 12, 10 and 9 clusters, respectively.

Figs. 3.7 (a), (b) and (c) show magnified CC clusters corresponding to Figs. 3.6 (a), (c) and (e), respectively. Let S_{CC} denote the cluster that includes the CC region. It is observed that, with $p = 0.08$ (see Fig. 3.6 (a)), S_{CC} roughly has the shape of the CC region, although there are some unconnected parts inside the CC region. With $p = 0.10$ (see Fig. 3.7 (b)), the S_{CC} follows the shape of the CC region more closely, and also the unconnected parts inside the CC region have been eliminated. On the other hand, with $p = 0.12$ (see Fig. 3.6 (c)), the S_{CC} cannot keep the rough shape of the CC region. In other words, $p = 0.10$ provides the best clustering performance.

Another example of cluster maps and histograms generated by AMS with $p = 0.08$, $p = 0.10$ and $p = 0.12$ are shown in Fig. 3.8. For this image also, $p = 0.10$ provides the best performance. Based on our experimental results, we select $p = 0.10$ to obtain a good initial cluster map generation for CC segmentation.

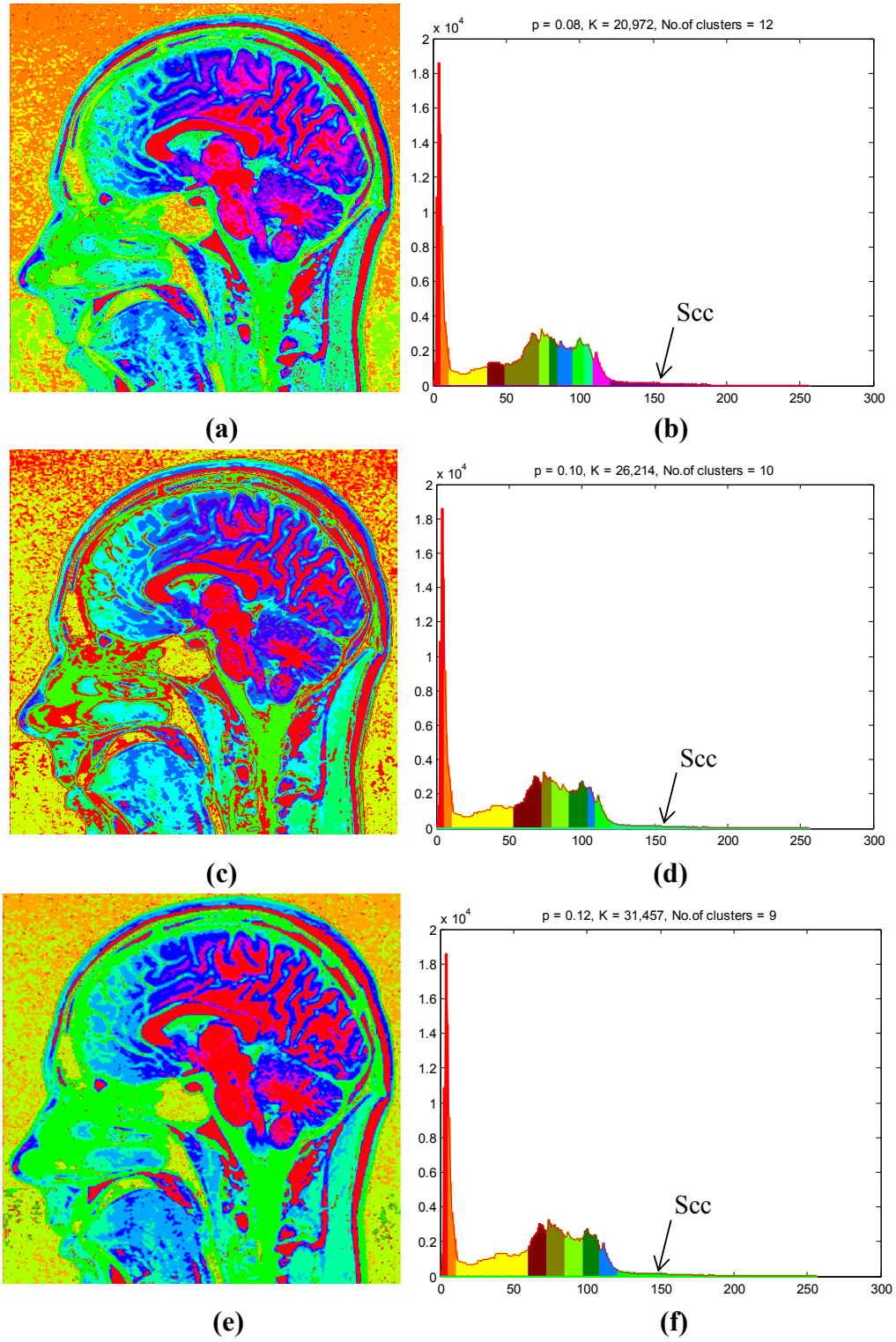
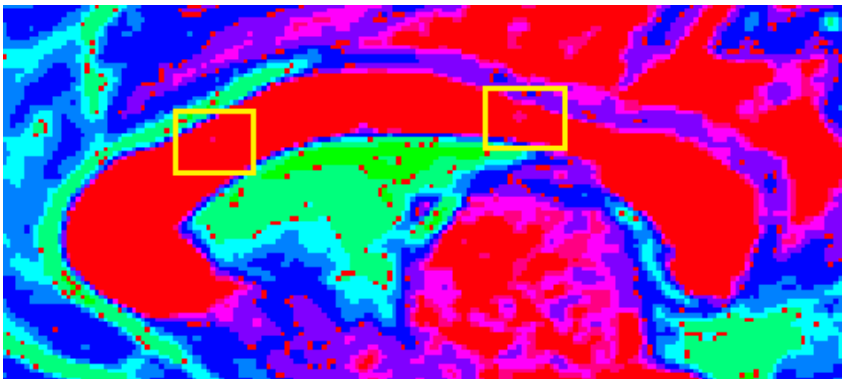
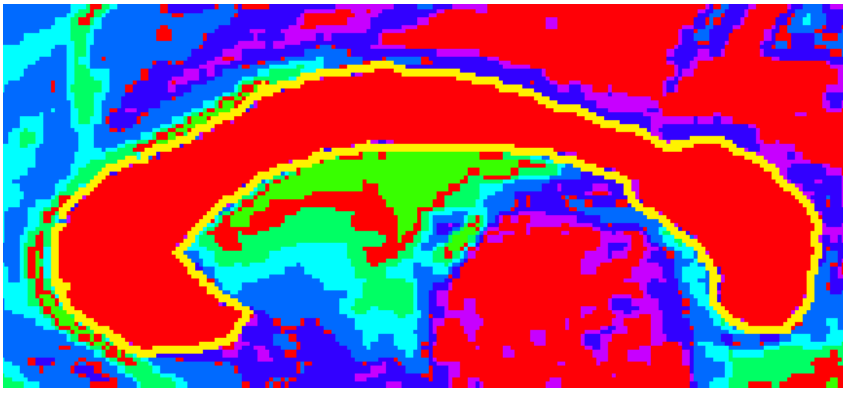


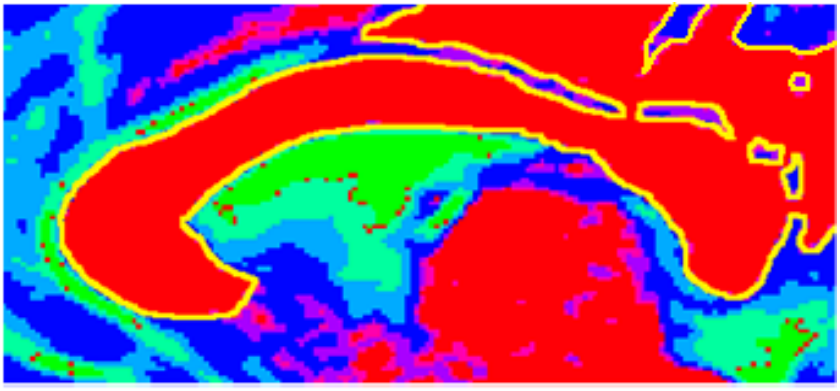
Fig. 3.6. Cluster maps and histograms generated by AMS. (a)(c)(e) Cluster maps generated by AMS with $p = 0.08$, $p = 0.10$, $p = 0.12$ respectively. (b)(d)(f) Cluster histograms corresponding to (a)(c)(e).



(a)



(b)



(c)

Fig. 3.7. Magnified CC Clusters. (a) is for Fig. 3.6 (a), and the yellow rectangles indicate the unconnected parts inside the CC, (b) is for Fig. 3.6 (c), and (c) is for Fig. 3.6 (e), and the yellow contour indicate the boundary of connected parts inside the CC and around the CC.

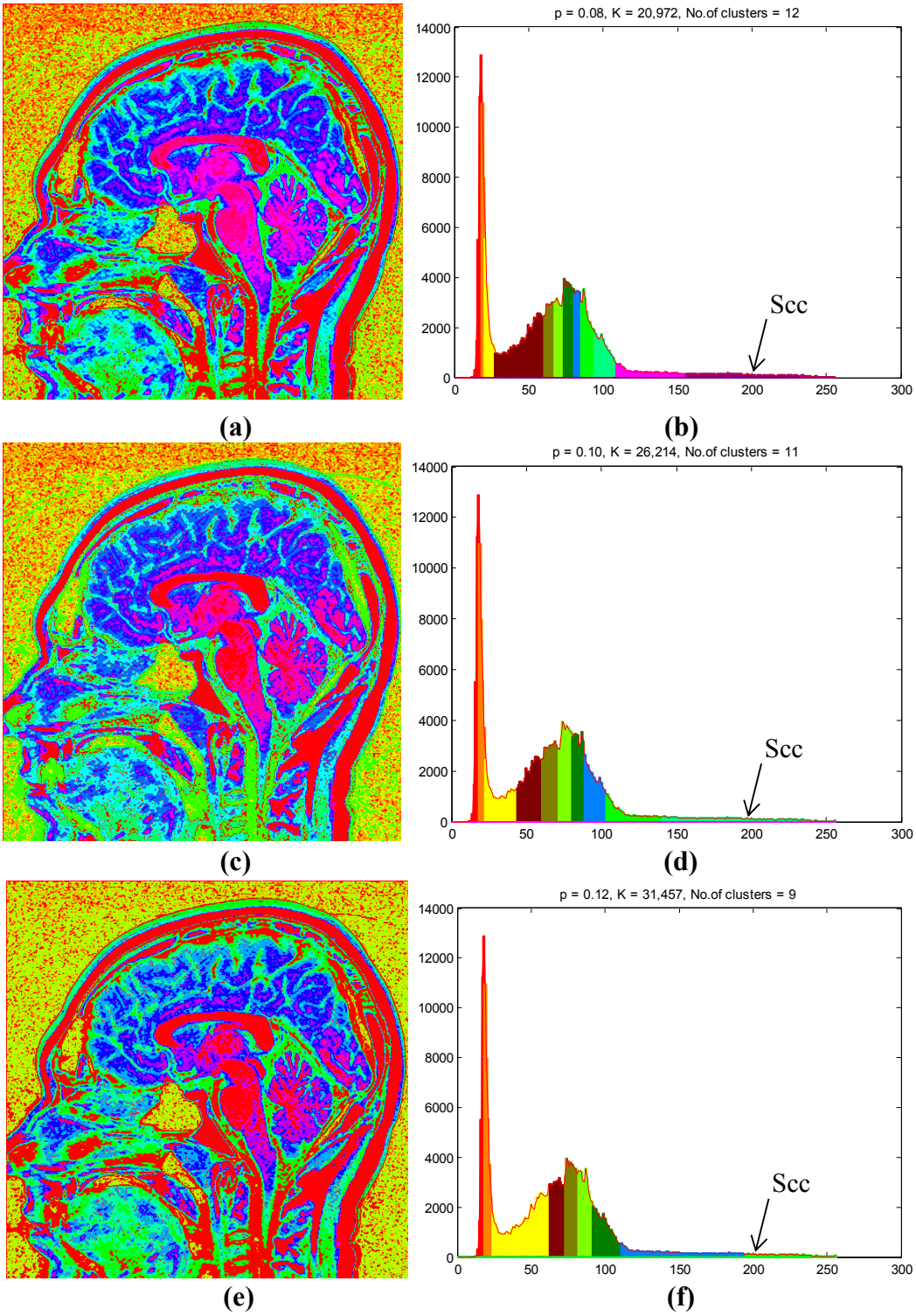


Fig. 3.8. Cluster maps and histograms generated by AMS. (a)(c)(e) Cluster maps generated by AMS with $p = 0.08$, $p = 0.10$, $p = 0.12$ respectively. (b)(d)(f) Cluster histograms corresponding to (a)(c)(e).

3.2.2 Automated CC Contour Initialization (ACI)

Given the cluster map generated by the AMS technique, three steps are followed to identify the cluster that contains the CC region. The boundary of the resulting cluster will be extracted and used as an initial contour in the subsequent GAC Model [16]. The details of the hybrid initialization technique are presented below.

3.2.2.1 Area Analysis

The CC is the largest white matter tissue in the brain. Therefore, we perform an area analysis to detect the candidate CC clusters from all clusters generated by the AMS technique. Based on the prior knowledge, the number of pixels that are included in the CC cluster is larger than a predefined threshold T_{Thr}. We set $T_{Thr} = 0.10N$ experimentally, where N is the total number of pixels in the brain MRI. The fraction 0.10 is determined based on the domain prior and experimental results. To be specific, let the i^{th} cluster obtained from AMS clustering be denoted by C_i, the candidate CC clusters be denoted by Cl_{Cand}, and the number of pixels included in a cluster be denoted by P_C. The parameter C_i is determined as follows:

$$C_i = \begin{cases} Cl_{Cand}, & if \quad P_C \geq T_{Thr} \\ others & otherwise \end{cases} \tag{3.8}$$

By applying the area criterion to the pre-clustered regions, we obtain some binary images with a few candidate regions. Denote the obtained binary image as I_B. Examples of area analysis results corresponding to Fig. 3.6 (c) are shown in Fig. 3.9,

where the white regions indicate the candidate cluster regions. Based on the observation of CC on brain MRI, the appearance of CC is much brighter than the appearance of background, in other words, the intensity of CC is much higher than the intensity of background. Therefore, only the cluster with higher intensity would be selected as a CC cluster.

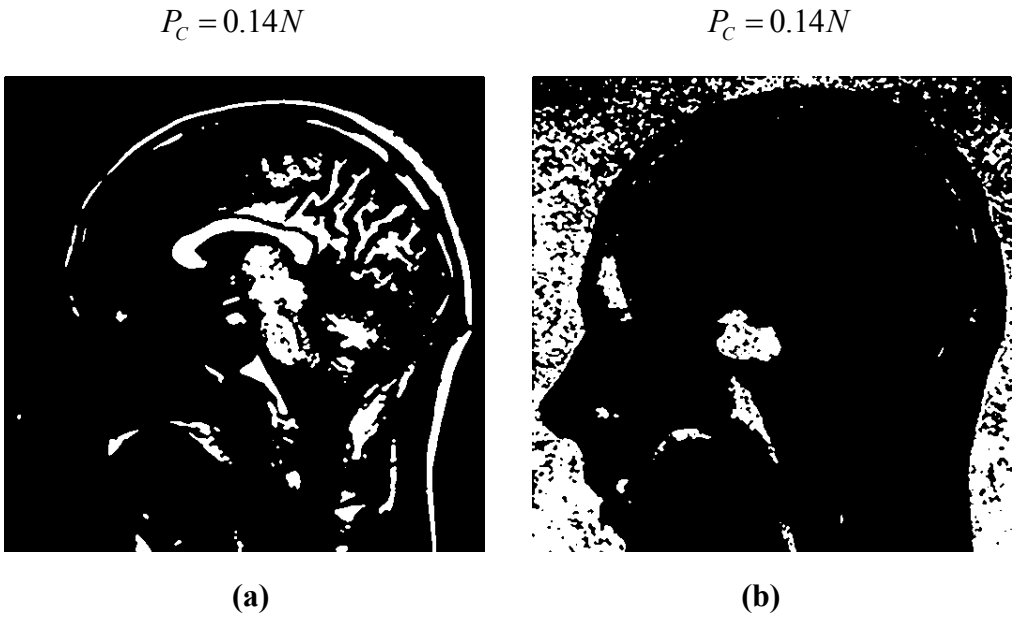


Fig. 3.9. Examples of area analysis results. (a) The largest cluster P_{10} with $P_C = 0.14N$ (having grayscale values in the range of $[122,256]$), (b) The second largest cluster P_2 with $P_C = 0.14N$ (having grayscale values in the range of $[6,11]$).

3.2.2.2 Template Matching

As shown in Fig. 3.9, besides cluster regions containing the object of interest, a few other cluster regions may exist. Therefore, in this step, the template matching (TM)

technique [21] is applied to detect the CC area. This step can be divided into several sub steps which are presented as follows:

Step 1: In the first step, a set of template images is generated based on a CC region obtained from a healthy person. Based on the knowledge of CC, a template image database is generated with templates of various geometric transformations (scaling, shearing, rotation). The scaling parameters $[S_x, S_y]$, rotation angle θ and shear transform parameters $[Sh_x, Sh_y]$ are incorporated using the following coordinate substitutions:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & -\cos\theta \end{bmatrix} \begin{bmatrix} 1 & Sh_y \\ Sh_x & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.9}$$

where x and y respectively denote the horizontal and vertical coordinates of a pixel. In our work, we have generated a template database with parameters: $S_x, S_y \in \{0.8, 0.9, 1\}$, the clockwise rotation angles $\theta = -15^\circ, 0^\circ, 15^\circ, 30^\circ$, and $Sh_x, Sh_y \in \{0, 0.05, 0.1, 0.15\}$. In other words, 48 template images are generated from a model CC template image, and all of the generated template images are shown in Fig. 3.10.

Step 2: Given the dataset of CC templates with various sizes, orientations and shapes, each CC template is translated to every possible location in the binary image I_B.

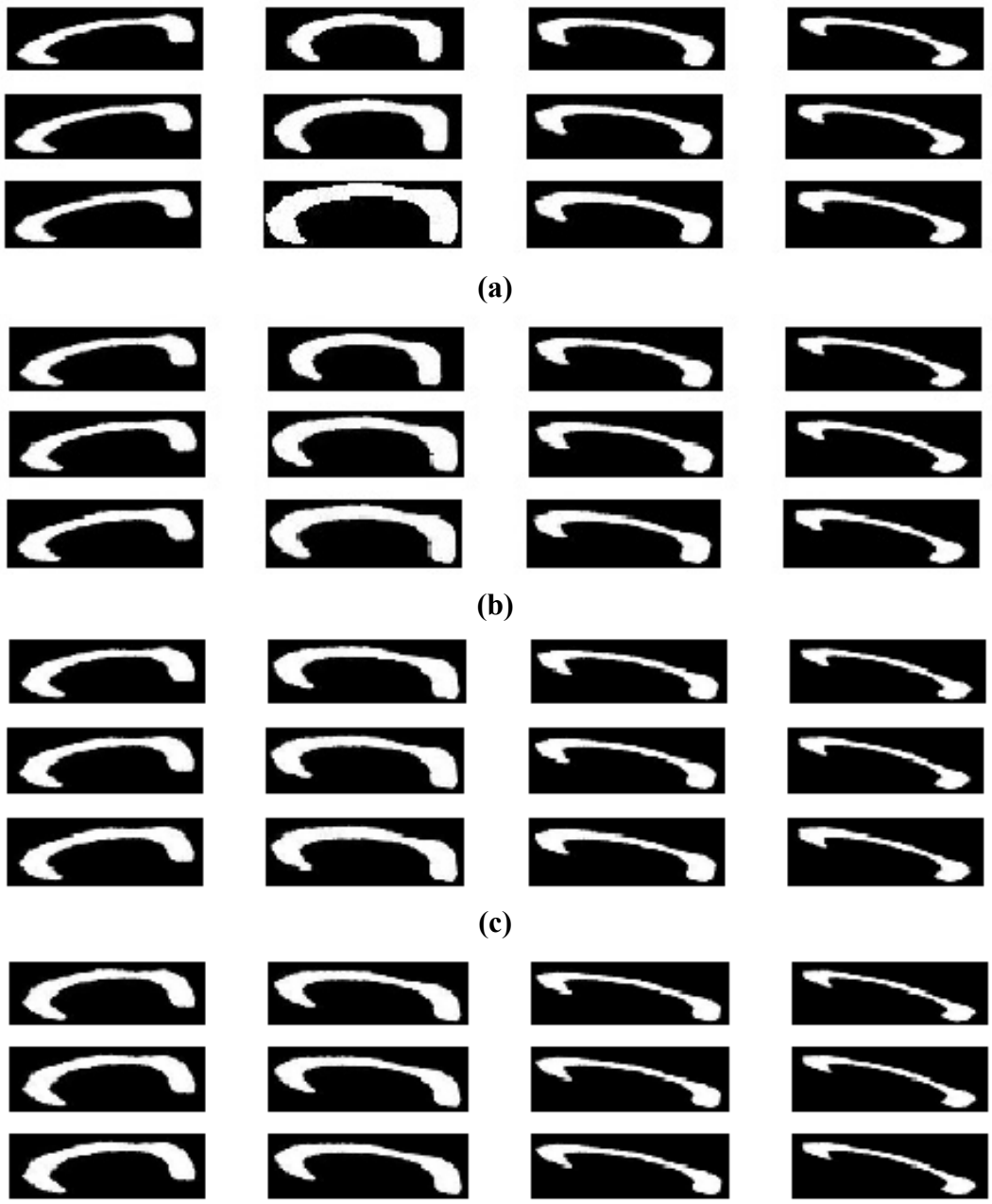


Fig. 3.10. Examples of CC templates with parameters $S_x, S_y \in \{0.8, 0.9, 1\}$, $\theta = -15^o, 0^o, 15^o, 30^o$, **(a)** $Sh_x, Sh_y \in \{0\}$, **(b)** $Sh_x, Sh_y \in \{0.05\}$, **(c)** $Sh_x, Sh_y \in \{0.1\}$, **(d)** $Sh_x, Sh_y \in \{0.15\}$.

The similarity between a template t and a sub-image f is measured by calculating the value of normalized cross correlation (NCC) [21], which is defined as:

$$\gamma(u,v)=\frac{\sum_{x,y}(f(x,y)-\overline{f}_{u,v})(t(x-u,y-v)-\bar{t})}{\sqrt{\sum_{x,y}(f(x,y)-\overline{f}_{u,v})^2\sum_{x,y}(t(x-u,y-v)-\bar{t})^2}} \tag{3.10}$$

where $f(x,y)$ is the pixel in the sub-image, (u,v) is the center of sub-image, $\overline{f}_{u,v}$ is the mean of sub-image, whereas $t(x-u,y-v)$ is the pixel in the template, and $\bar{t}$ is the mean of template. The value of NCC is between 0 and 1, and higher value indicates higher similarity. By translating each template on the image I_B, we obtain the response in mage with NCC values.

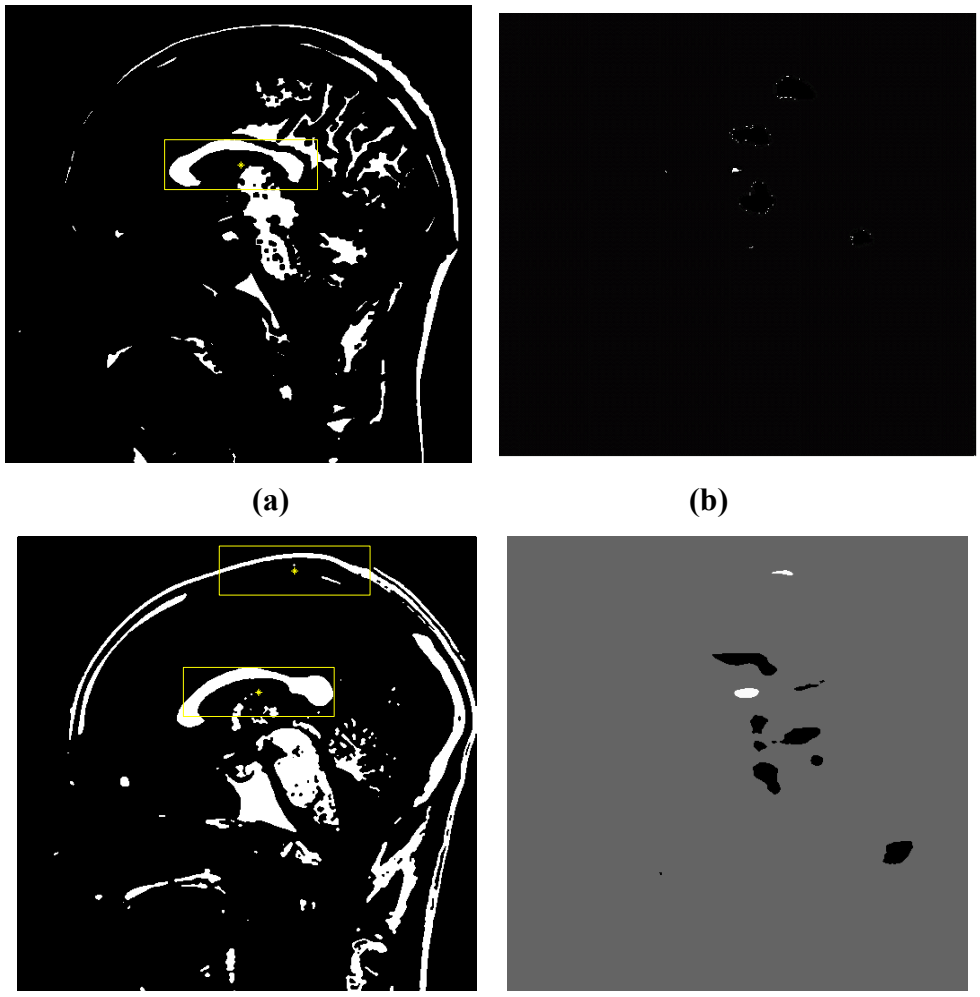
Step 3: A block centered at (u,v) is a candidate of CC, when satisfying the following condition:

$$\gamma(u,v)\geq T_{NCC} \tag{3.11}$$

where T_{NCC} is a predefined threshold. In our work, T_{NCC} is empirically set to 0.7.

Examples of two template matching results are shown in Fig. 3.11 (a) and Fig. 3.11 (c) respectively, and the corresponding NCC images are shown in Fig. 3.11 (b) and Fig. 3.11 (d) respectively. The brightest part reflects regions with $\gamma\geq 0.7$, the gray part reflects regions with $0.4<\gamma<0.7$, and the black part reflects regions with $\gamma\leq 0.4$. It is noticeable that a spurious candidate cluster that contains the top skull has been shown near the top of Fig. 3.11 (c). The reason is that the top skull has similar appearance of CC, when the template matching technique is applied to the

cluster containing the top skull, the top skull would be easily identified as a candidate cluster region.



(a) **(b)**

(c) **(d)**

Fig. 3.11. Examples of template matching results. (a)(c) Two template matching results, and the yellow rectangles indicate the detected regions. (b)(d) The corresponding NCC images of (a)(c).

3.2.2.3 Shape and Location Analysis

As shown in Fig. 3.11(c), besides cluster regions containing the object of interest, spurious CC clusters may be observed. Therefore, in this step, the supplementary

analysis is required to further improve the initialization performance. In the proposed technique, location analysis of candidate regions are performed, and several sub steps are presented as follows:

Step 1: Note that the CC region is usually located closer to the center of the brain MR images, compare with the top skull. In order to ignore the effect of image size, the location feature can be defined by the distance (l_d) between candidate cluster region center and the image center to the image height (l_h) ratio.

Step 2: For a candidate cluster region, we use the location criteria based on the prior domain knowledge to determine whether it is a CC region or spurious CC region.

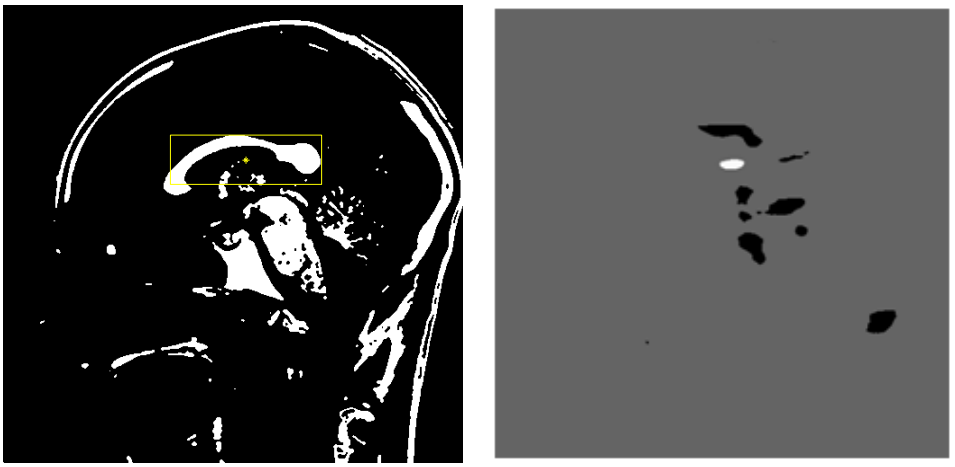
$$A = \begin{cases} CC & if\ l_d / l_h \le T_L \\ spurious\ CC & otherwise \end{cases} \tag{3.12}$$

Threshold T_L is set to 0.25 empirically to make sure that the CC region is located near the image center. Furthermore, extract the largest connected area A (with largest number of pixels) in the detected region. An example of shape and location analysis result has been shown in Fig. 3.12. It is noticeable that the spurious candidate cluster as shown in Fig. 3.11 (c) has been filtered out.

Step 3: The boundary of the CC area detected in step 2 is extracted as the initial contour of CC. Two examples are illustrated in Fig. 3.13.

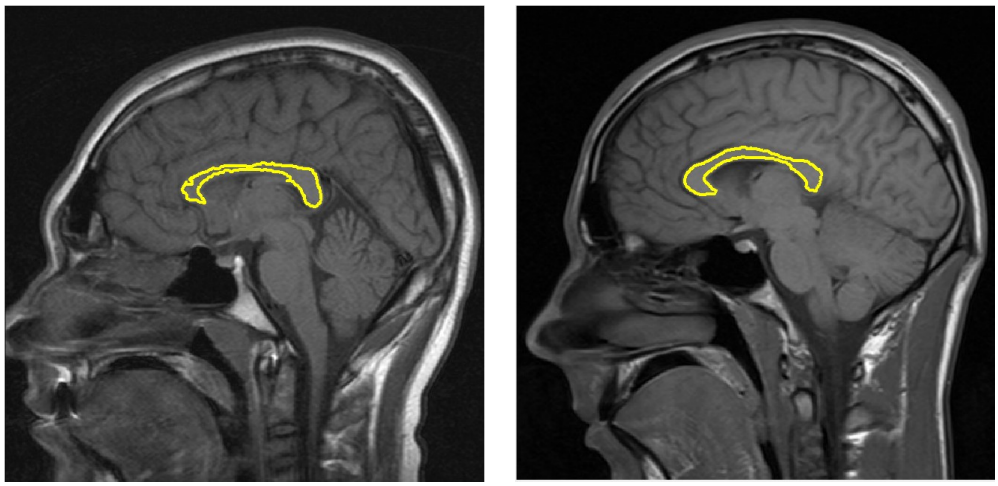
3.2.3 Geometric Active Contour Based Segmentation

Geometric active contour is a curve evolution technique that implicitly represents the curve as the zero level set of a high dimensional surface. In the GAC



(a) **(b)**

Fig. 3.12. An example of shape and location analysis result, the yellow rectangles indicate detected regions. (b) The corresponding NCC images of (a).



(a) **(b)**

Fig. 3.13. Examples of initial contour of CC, the yellow contour indicates the generated initial contour of CC.

framework, active contours are deformable curves that can be used to delineate the image structures. After the initialization of contour, the contour is allowed to evolve to minimize an image energy functional, and it has been extensively used in image segmentation [22-25]. Although the computation in the GAC framework is relatively expensive, the GAC framework can handle complex topology changes, so it would be appropriate to use the GAC-based technique to segment CC.

In the GAC technique [16], the contour evolves by maximizing the distance between the intensity probability density functions of interior region and exterior region of contour. By doing this, the statistical information of image regions has been incorporated into the GAC framework. The schematic of the typical GAC-based CC segmentation technique is shown in Fig. 3.14.

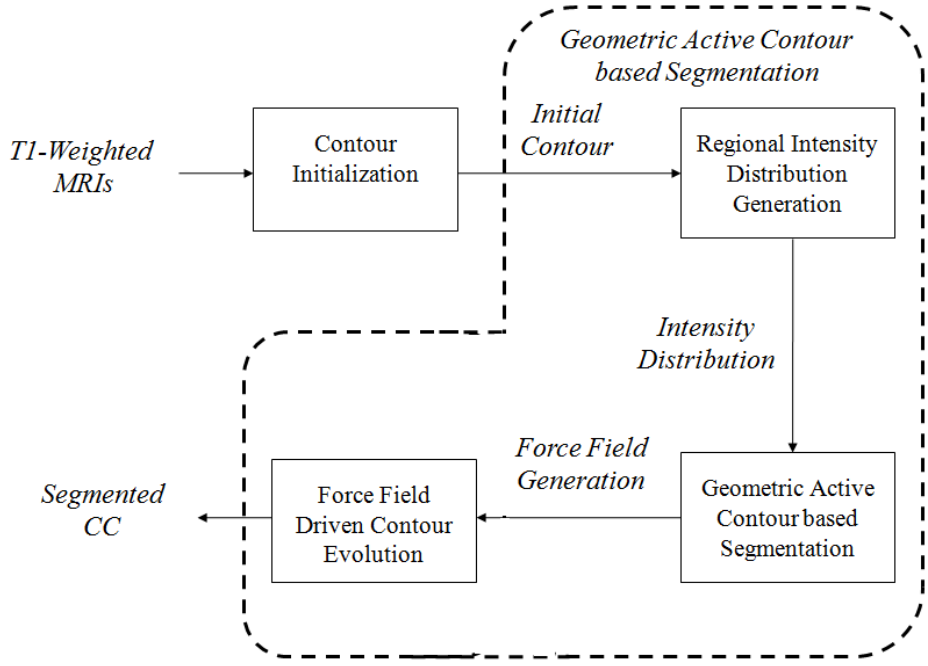


Fig. 3.14. Simplified schematic of the typical GAC-based CC segmentation

In section 3.2.2, the initial contour of CC area is generated using a proposed AMS-ACI technique, and this close-to-target initialization will facilitate the quick

evolution of contour in the GAC-based technique [16] for CC segmentation. In this section, the final segmentation result of CC area is captured by the GAC technique [16].

Let C denotes the initial contour of CC area, and C represents a set of zeros of a signed distance function $\phi : \Re^2 \to \Re$, let the input image be defined on the domain Ω, i.e.,

$$\begin{cases} C = \{x \in \Omega : \phi(x) = 0\} \\ \text{int}\,erior(C) = \{x \in \Omega : \phi(x) < 0\} \\ exterior(C) = \{x \in \Omega : \phi(x) > 0\} \end{cases} \tag{3.13}$$

The initial contour C evolves in the hope that the interior region of C matches the CC region, and the exterior region of C matches the background. In the GAC-based technique [16], the image intensity probability density functions of the background and object are incorporated into the process of contour evolution.

Considering that the gray scale intensities of the image are modeled as the random variable $z \in Z$. Therefore, the *pdf*s $p_{in}(z,\phi)$ and $p_{out}(z,\phi)$ of intensities in the interior and exterior of the contour C can be evaluated respectively.

The similarity between intensity distributions in the interior and exterior of the contour C is measured by the standard deviation between the log-likelihood of $p_{in}(z,\phi)$ and $p_{out}(z,\phi)$. In the GAC model [16], it is assumed that there is a distinct difference between the intensity distribution in the interior and exterior of the contour. Based on this assumption, the evolution of the contour would be driven in finding the maximum similarity between $p_{in}(z,\phi)$ and $p_{out}(z,\phi)$. To maximize the

above similarity, the contour C is evolved iteratively using the image energy function $E(z,\phi)$ defined as follows:

$$E(z,\phi)=\sqrt{\varepsilon\left[\left(\log\frac{p_{in}(z,\phi)}{p_{out}(z,\phi)}\right)^2\right]-\varepsilon\left[\left(\log\frac{p_{in}(z,\phi)}{p_{out}(z,\phi)}\right)\right]^2} \tag{3.14}$$

where $\varepsilon[f(z)]$ denotes the expected value of the functional $f(z)$. In the GAC model [16], the value of z is restricted to a set of grayscale intensity values $\{1,2,\ldots,256\}$. $p_{in}(z,\phi)$ and $p_{out}(z,\phi)$ are the *pdf*s defined on the variable z. The evolution of contour C (or equivalent ϕ) is performed according to the equation,

$$\frac{\partial\phi}{\partial t}=\nabla_{\phi}E(z,\phi) \tag{3.15}$$

The evolution is stopped if the maximum number of iterations has been achieved.

3.3 Summary

In this chapter, a novel technique for automated segmentation of CC in T1-weighted misdagittal brain MRIs was proposed. In the proposed technique, the initial brain MRI is first clustered into various homogeneous regions using the adaptive mean shift technique, representing various brain tissues. Region analysis, template matching, combined with the shape and location analysis techniques are used to localize the CC region from the clusters generated by the adaptive mean shift technique. The boundary of obtained CC region is extracted as the initial contour of the subsequent deformation model. Finally, the segmentation of CC is generated using the GAC model.

Chapter 4

Graphical User Interface Design and Performance Evaluation

In this chapter, we first present a graphical user interface (GUI), which has been designed for the proposed CC segmentation technique (see Chapter 3). The performance evaluation of automated initial CC contour estimation and the performance of automated segmentation are then presented.

4.1 Introduction

The GUI is the type of user interface that helps users communicate with software and electronic devices in a graphical display mode. Compared with command-line interfaces, GUI has advantages in usability and user experience. The goal of GUI design is to meet the exact requirements of users, and make sure that the communication between users and software / electronic devices as simple and effective as possible. In order to facilitate doctors to use the proposed technique on

brain MR images, we developed a simple and effective GUI in MATLAB. The details of the GUI development are presented in Section 4.2.

4.2 Design of Graphic User Interface

In this section, we introduce the process of GUI design, that is based on MATLAB GUIDE (GUI development environment). Considering that the goal of our GUI design is simplicity, on the main panel, we only place an aexs object, that is used to display an image, as well as three choices, which are Load Image, Generate Initial Contour, and CC Seg. The designed GUI provides a clear look, that largely avoids screen clutter. Figure 4.1 shows the main panel of the designed MATLAB GUI. By clicking the "Load Image" button, we can search and load an existing brain MR image, and it will be displayed in the axes object on main panel, as shown in Fig. 4.2. After selecting an brain MR image, by clicking the "Generate Initial Contour" button, the input image will be processed, and the generated initial contou will be shown as the yellow contour on the input image. Figure 4.3 illustrates an example of the generated initial contour showing on the input brain MR image. By clicking the "CC Seg" button, the final contour of CC will be estimated and displayed as the yellow contour on the input image, as shown in Fig. 4.4.

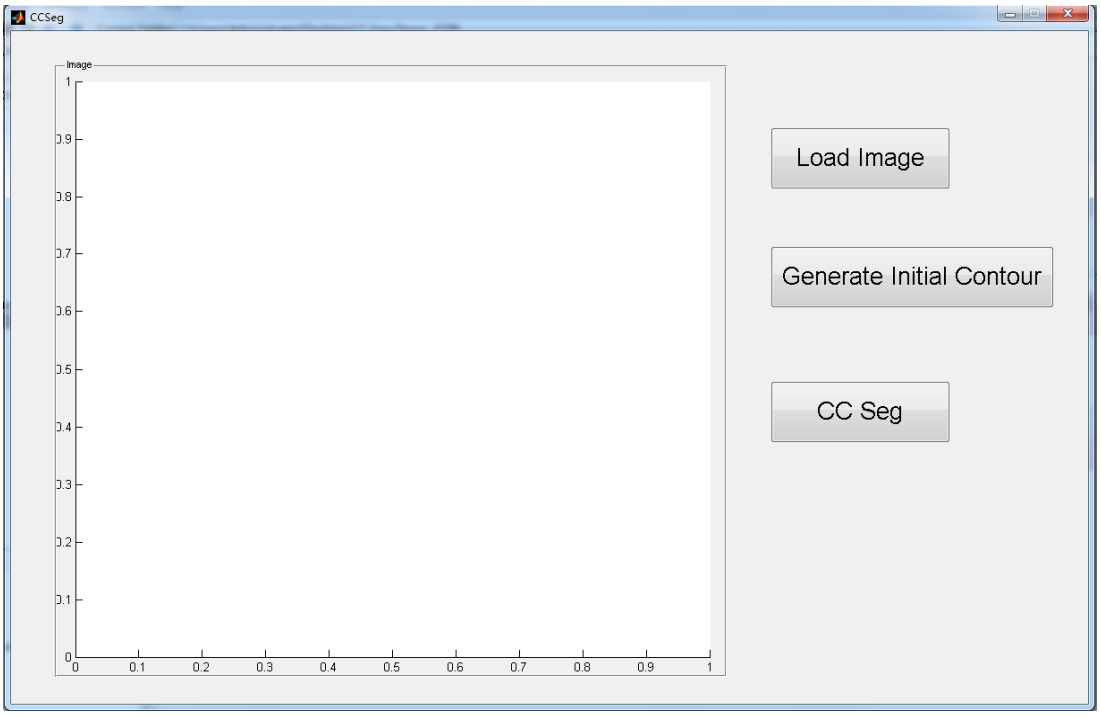


Fig. 4.1. Main panel of the designed MATLAB GUI

Fig. 4.2. Uploading of a brain MR image

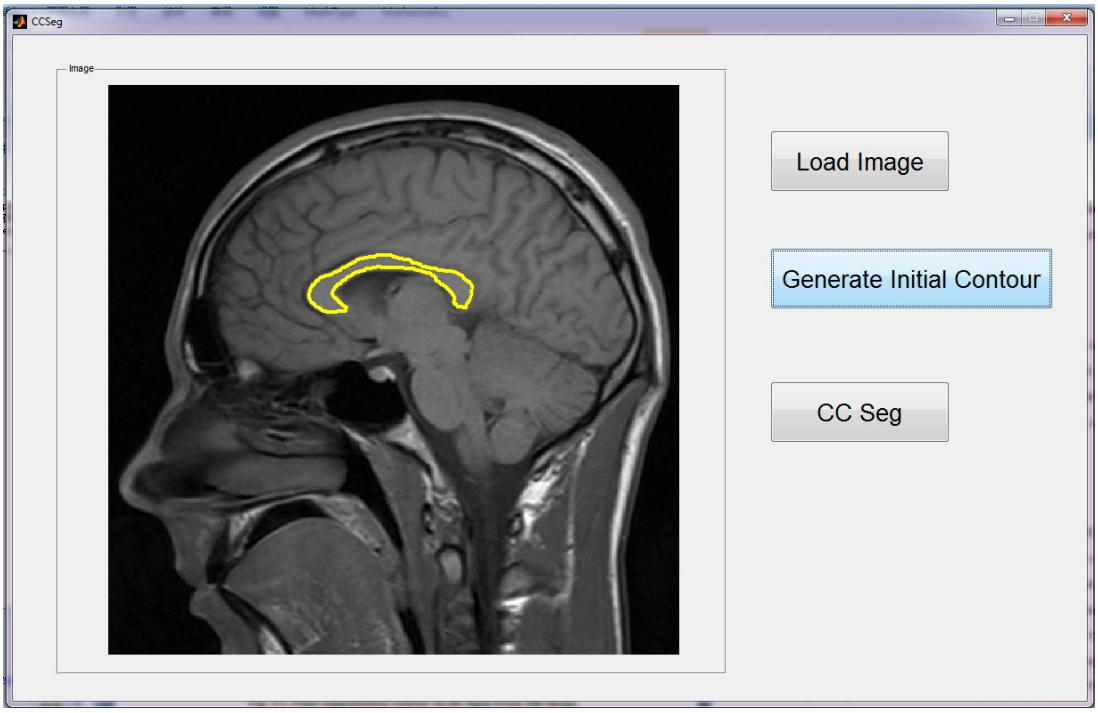


Fig. 4.3. Generating initial contour on the input brain MR image

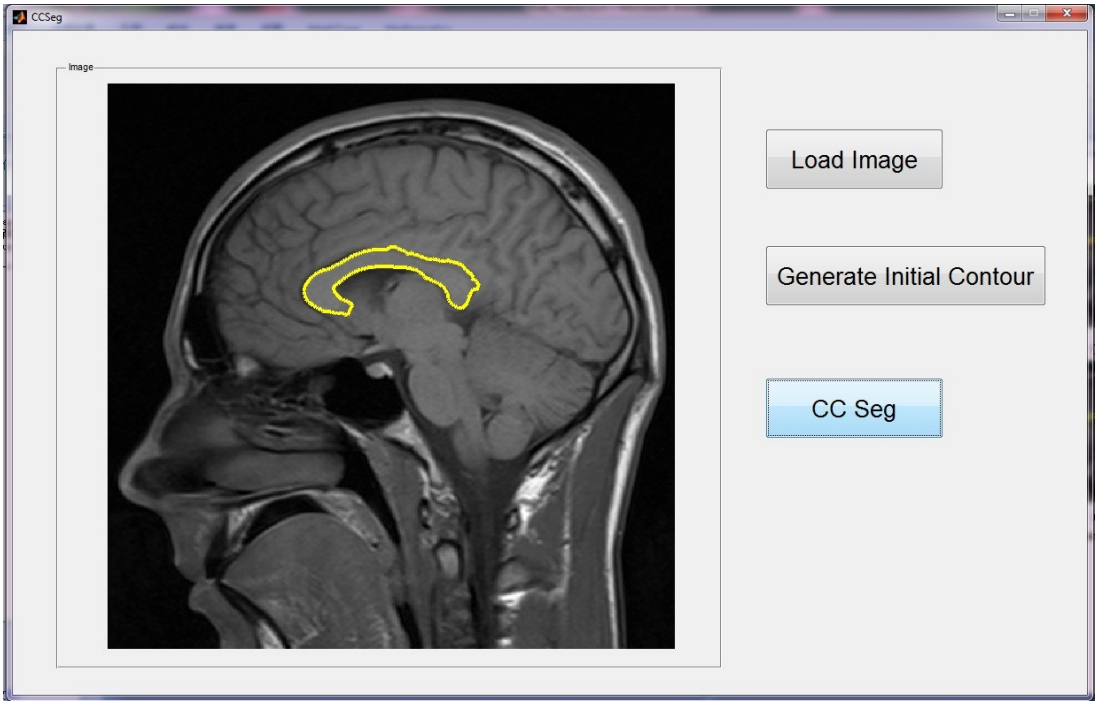


Fig. 4.4. Final segmentation contour on the input brain MR image

4.3 Performance Evaluation

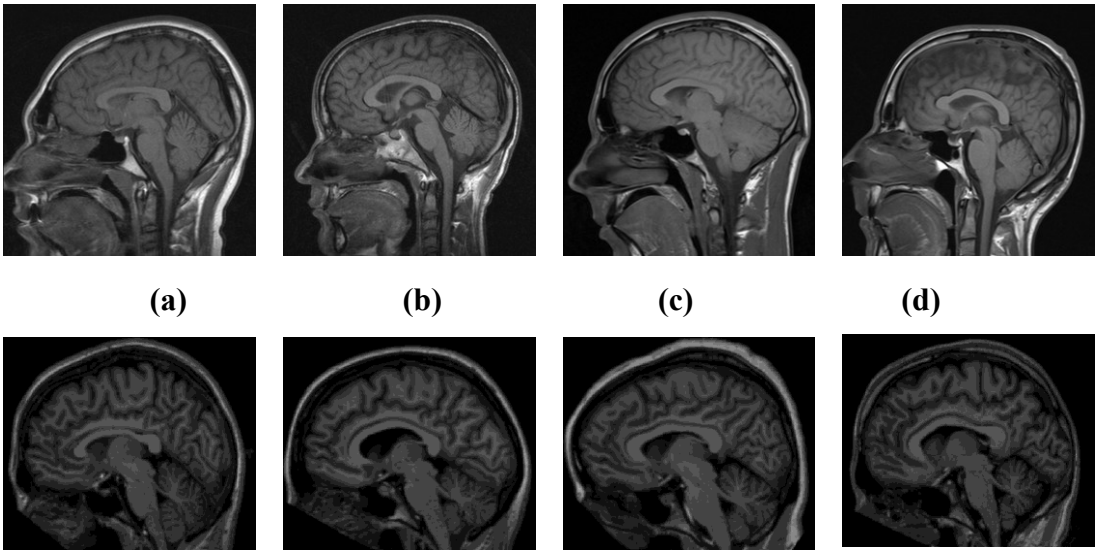
In this section, we present the performance evaluation of proposed AMS-ACI technique, AMS-ACI-GAC segmentation technique and typical GAC technique [16] on brain MRI database.

4.3.1 Database Used

The brain MRI data were chosen from two brain MRI databases: 1) brain MRI (BMRI) database from the University of Alberta Hospital; 2) a publicly available Open Access Series of Imaging Studies (OASIS) MRI database [26]. The OASIS MRI database has been commonly used in the study of CC for normal people and patients with brain disorders [11][28]. The BMRI database contains 7 images from patients with neurological disorders, and the OASIS database contains 27 normal images from subjects in the age range 20 to 40. In Fig. 4.5, examples of images of BMRI database and OASIS database are shown in Fig. 4.5 (a)-(d) and Fig. 4.5 (e)-(h) respectively.

4.3.2 Evaluation Metrics

The ground truths of CC regions of images in the BMRI database are manually drawn by a professional neurologist from the University of Alberta Hospital. The ground truths of CC regions of images in the OASIS database are manually drawn according to medical knowledge learnt from the professional neurologist. The segmentation results are compared with the ground truths.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4.5. Examples of images used in the performance evaluation. (a)-(d) Images of BMRI database; (e)-(h) Images of OASIS database.

To evaluate the segmentation performance, *Accuracy*, *Sensitivity*, *F1-score* and *Time-cost* are considered as evaluation metrics. The calculation of *Accuracy*, *Sensitivity* and *F1-score* are defined as follows [28]:

$$Accuracy = \frac{TP}{TP + FP} \tag{4.1}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.2}$$

$$F1 = \frac{2\,AccSen}{Acc + Sen} = \frac{2TP}{2TP + FP + FN} \tag{4.3}$$

where TP is the number of pixels in the true positive area (image region which is correctly classified as CC), TN is the number of pixels in the true negative area (image region which is correctly classified as background), FP is the number of pixels in the false positive area (image region which is incorrectly classified as CC),

and *FN* is the number of pixels in the false negative area (image region which is incorrectly classified as background). The illustration of *TP* (in blue), *TN* (in gray), *FP* (in yellow), *FN* (in green) is shown in Fig. 4.6. The *time-cost* is defined as the running time of generating segmentation result based on the same image. The experimental platform is Matlab R2011a on a computer with Intel Core i7-3667U CPU 2GHz and 8GB RAM.

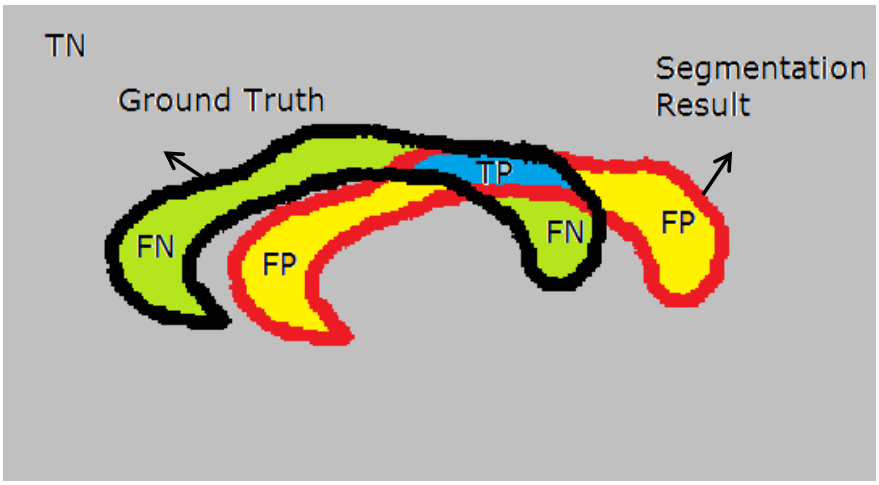
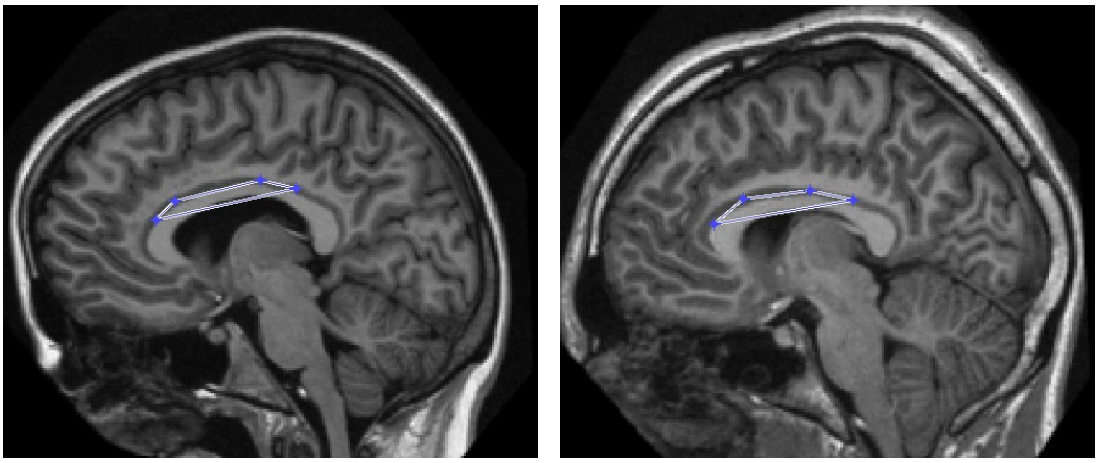


Fig. 4.6. The distribution of *TP* (in blue), *TN* (in gray), *FP* (in yellow), *FN* (in green). The black contour indicates the ground truth, and the red contour indicates the segmentation result.

4.3.3 Experiments and Analysis

Given the BMRI and OASIS database, we compare the segmentation performance of the proposed AMS-ACI technique, AMS-ACI-GAC segmentation technique, and the typical GAC technique [16]. The initial contour of typical GAC technique [16] is a quadrilateral that is manually placed on the location of CC. In the experiment, the principle of initial contour design is that the contour would include the CC region as

much as possible and include surrounding regions as less as possible. Two examples of initial contour of typical GAC technique [16] are shown in Fig. 4.7.



(a) (b)

Fig. 4.7. Initial Contours of typical GAC technique [16] in the experiments

The evaluation results are shown in Table 4.1. *Accuracy*, *Sensitivity*, *F1-score* and *Time-cost* of different segmentation techniques using both BMRI and OASIS databases are compared. The values of *Accuracy*, *Sensitivity*, *F1-score* and *Time-cost* are represented by the average value ± standard deviation. It is noticeable that the average accuracy of the AMS-ACI technique is higher than the other two segmentation techniques. For example, while the average accuracy using the AMS-ACI technique is 98%, the accuracies of the AMS-ACI-GAC technique and typical GAC technique [16] are 95% and 86% respectively. In other words, the AMS-ACI is able to provide a close-to-target initial contour. Meanwhile, the AMS-ACI-GAC technique outperforms the typical GAC technique [16] in average F1-score. For example, while the average F1-score of CC segmentation is 85% using the typical GAC technique [16], the F1-score of the AMS-ACI-GAC technique is 88%, which

indicate that the proposed technique has higher true positive rate. As for the time-cost, typical GAC technique [16] is about 4 times than the proposed technique, and is about 40 times than the AMS-ACI technique, which indicates that the AMS-ACI technique and the proposed segmentation technique are more efficient for real-time operation.

Table 4.1. Evaluation of different segmentation techniques using the same database

Segmentation Techniques	Accuracy	Sensitivity	F1	Time Cost (s)
AMS-ACI	0.98 ± 0.04	0.66 ± 0.09	0.79 ± 0.06	2.27 ± 0.53
AMS-ACI-GAC	0.95 ± 0.09	0.84 ± 0.08	0.88 ± 0.04	24.72 ± 1.69
GAC [16]	0.86 ± 0.16	0.83 ± 0.06	0.85 ± 0.08	82.43 ± 11.27

Fig. 4.8 presents examples of comparison between ground truth and segmentation results based on AMS-ACI technique, AMS-ACI-GAC technique and typical GAC technique [16]. It can be observed that the AMS-ACI provides close-to-target initial contour, and the AMS-ACI-GAC technique gives more accurate results with respect to the ground truth.

Ground Truth GAC [16] AMS-ACI AMS-ACI-GAC

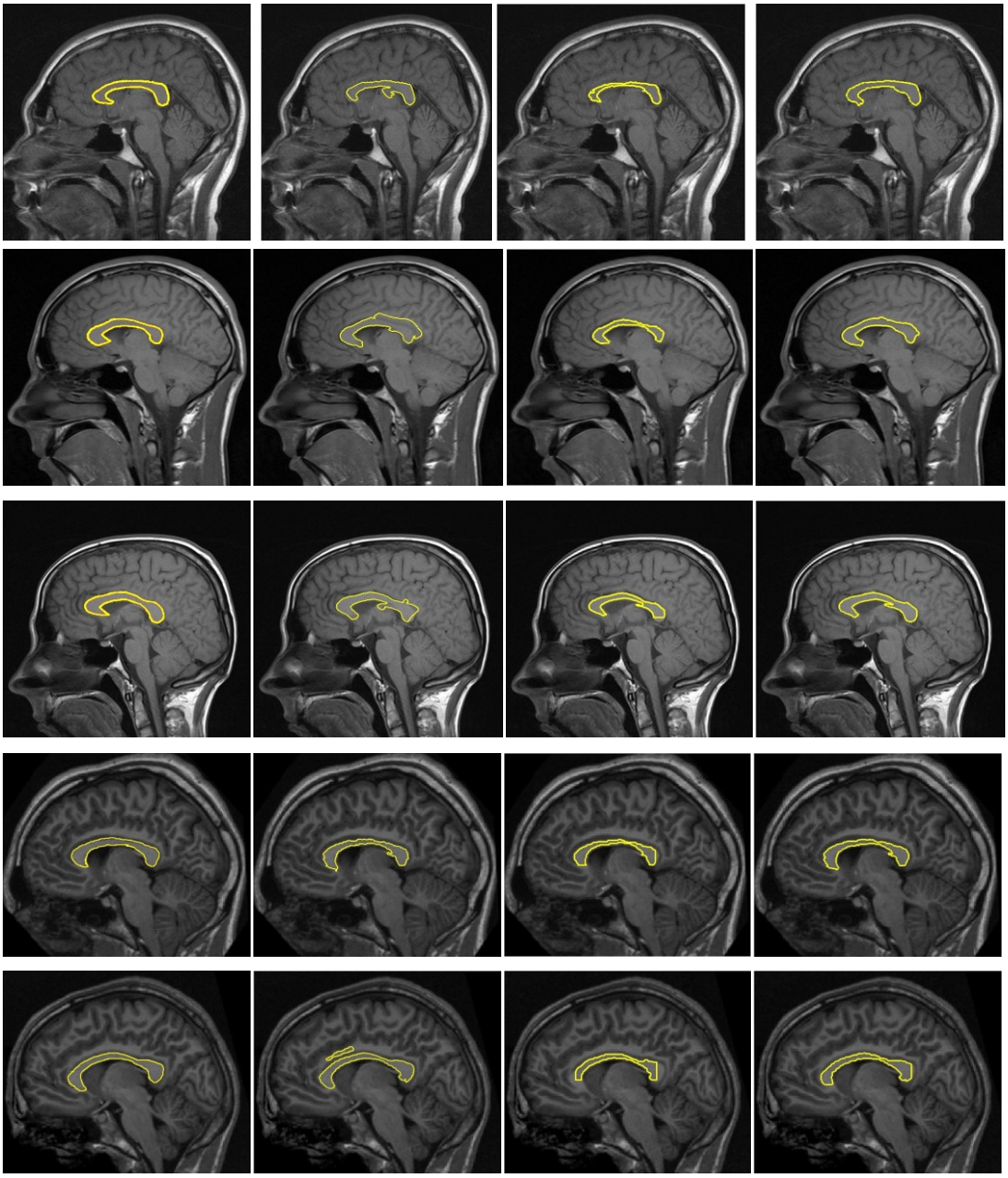


Fig. 4.8. Comparisons of segmentation results (yellow contour) between ground truth, typical GAC [16], proposed AMS-ACI and AMS-ACI-GAC on T1-weighted midsagittal brain MR images.

4.4 Summary

In this chapter, the development of a graphical user interface for the proposed CC segmentation technique has been presented. The performance evaluation results of AMS-ACI technique, AMS-ACI-GAC technique and typical GAC technique have also been provided.

Chapter 5

Conclusions and Future Work

Considering that CC region can be easily identified on visual inspection, and it is always adopted as the landmark of human brain. We therefore decided to develop and implement a CAD technique to identify this structure.

In this thesis, we have proposed an intelligent CAD system as a tool for the automated segmentation of CC in T1-weighted MR images. The proposed CAD system can be applied to patients with epilepsy and is expected to provide useful information for physicians in increasing diagnosis sensitivity, and it will also provide the ground work for utilizing a similar technique to identify abnormal brain structures with epileptogenic potential.

5.1 Conclusions

In the proposed CAD system, we use the midsagittal plane of T1-weighted brain MR images, where the CC region could be best viewed. As the general schematic of the proposed CAD system shown in Fig. 3.1, several homogeneous regions are first

generated using an AMS clustering technique. A hybrid initialization technique including area analysis, template matching, shape and location analysis is then used to identify the CC region, and the boundary of CC region is extracted as the initial CC contour. Finally, the close-to-target initial CC contour is used in the GAC-based technique [16] that captures the final segmentation of the CC region. Experimental results show that in the proposed system, the initial contour is quite close-to-target, and it is extremely helpful in quick convergence of the GAC-based technique [16], and the GAC-based technique [16] is successful in capturing the accurate segmentation result of CC region. Thus, an automated system for segmentation of CC region in T1-weighted midsagittal brian MR images is developed.

5.2 Future Works

Although the proposed CAD system for the automated segmentation of CC in T1-weighted midsagittal MR images shows promising results, there are more research work that could be done to improve the efficiency of the proposed CAD system. A few potential future research directions are as follows.

- In the preprocessing step, only the grayscale intensity feature of the brain MR images is used in the AMS clustering technique as this feature has been demonstrated to be efficient to generate a rough shape of CC region. However, other features, e.g., spatial features of brain MR images, may provide complementary useful information for the clustering of CC region.
- In this work, a CAD system that can segment the CC region from T1-weighted midsagittal brain MR images has been presented. The distinction

between patients with epilepsy and normal people will be a more challenging research work. By achieving this goal, detection and analysis of abnormality in CC region is required.

- In this work, the automatic segmentation of CC region has been investigated, which generally has clear margins. While detecting brain lesions, for example, brain tumors, sometimes it would be difficult to identify clear margins between lesions and neighboring normal brain tissues. Hybrid Fuzzy classification techniques could be applied to generate classes of normal and abnormal MRI slices [38-40].

5.3 Publications

The proposed technique has been published (or submitted) as follows:

- Yue Li, Mrinal Mandal, and S. Nizam Ahmed. "Fully automated segmentation of corpus callosum in midsagittal brain MRIs." *In Proc: The 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2013)*. August 2013, pp. 5111-5114.
- Yue Li, Nizam Ahmed, Mrinal Mandal. “An Efficient Technique for Corpus Callosum Segmentation in Midsagittal Brain MR Images,” submitted to *ICTACT Journal on Image and Video.*

Bibliography

[1] World Health Organization, “Computer-aided detection for Tuberculosis”, website, Apr. 2012. [Online]. Available: http://www.who.int/ehealth/resources/compendium_ehealth2012_4.pdf.

[2] Ville-Marie Medical Centre, “Full Field Digital Mammography with Computer Aided Detection (CAD)”, website, Jul. 2012. [Online]. Available: http://vmmed.com/montreal-mri/digital-mammography-cad/.

[3] Epilepsy Canada, “Epilepsy Facts”, website, Jan. 2014. [Online]. Available: http://www.epilepsy.ca/en-ca/facts/epilepsy-facts.html.

[4] Live Science, "What is an MRI (Magnetic Resonance Imaging)?" website, Dec. 2014. [Online]. Available: http://www.livescience.com/39074-what-is-an-mri.html.

[5] H. T. Chugani, *Neuroimaging in Epilepsy*. Oxford University Press, 2011.

[6] R. O'Dwyer, et al., “Differences in corpus callosum volume and diffusivity between temporal and frontal lobe epilepsy”, *Epilepsy & Behavior*, vol. 19, no. 3, pp. 376-382, 2010.

[7] H. Chahboune, et al. “DTI abnormalities in anterior corpus callosum of rats with spike-wave epilepsy”, Neuroimage, vol. 47, no. 2, pp. 459-466, 2009.

[8] E. V. Sullivan, T. Rohlfing, and A. Pfefferbaum. "Longitudinal study of callosal microstructure in the normal adult aging brain using quantitative DTI

fiber tracking." Developmental neuropsychology, vol.35, no.3, pp.233-256, 2010.

[9] S. A. Kozlovskiy, A. V. Vartanov, M. M. Pyasik, E. Y. Nikonova. "Functional role of corpus callosum regions in human memory functioning". International Journal of Psychophysiology. vol.85, no.3, pp.396–397, 2012.

[10] M. Ingalhalikar, et al. "Sex differences in the structural connectome of the human brain." Proceedings of the National Academy of Sciences vol.111, no.2, pp.823-828, 2014.

[11] B. A. Ardekani, K. Figarsky, J. J. Sidtis. "Sexual Dimorphism in the Human Corpus Callosum: An MRI Study Using the OASIS Brain database". Cerebral Cortex. vol.23, no.10, pp.2514–2520, 2012.

[12] DNA Illustrations, Inc., "Portfolio Gallery", website, Jun. 2016. [Online]. Available: http://dnaillustrations.com/inside-art.html.

[13] M. Brejl and M. Sonka, "Object localization and border detection criteria design in edge-based image segmentation: automated learning from examples," *IEEE Trans. on Medical Imaging*, vol. 19, no.10, pp. 973-985, Oct. 2000.

[14] B. V. Ginneken, A. F. Frangi, J. J. Staal, B. M. T. H. Romeny, and M. A. Viergever, "Active shape model segmentation with optimal features," IEEE Trans. on Medical Imaging, vol. 21, no.8, pp.924-933, Aug. 2002.

[15] M. Jacob, T. Blu, and M. Unser, "Efficient Energies and Algorithms for Parametric Snakes," IEEE Trans. on Image Processing, vol. 13, no.9, pp.1231-1244, Sep. 2004.

[16] R. Sandhu, T. Georgiou, and A. Tannenbaum, "A new distribution metric for image segmentation", in Proc. SPIE'08, vol. 691404, pp. 1-9, 2008.

[17] J. Zhou, S. Chang, S. Zhang, G. Pappas, M. Michaelides, F. Delis, N. Volkow, P. Thanos, and D. Metaxas, "A novel learning based segmentation method for rodent brain structures using MRI," in Proc. ISBI'08, 2008, pp. 61-64.

[18] N. El-Zehiry, M. Casanova, and A. Elmaghraby, "Variability of the relative corpus callosum cross sectional area between dyslexic and normally developed brains," in Proc. ISBI'08, 2008, pp. 436-439.

[19] R. Lai, Y. Shi, N. Sicotte, and A. W. Toga, "Automated corpus callosum extraction via Laplace-Beltrami Nodal Parcellation and Intrinsic Geodesic Curvature Flows on surfaces," in Proc. ICCV'11, 2011, pp. 2034-2040.

[20] S. Anand, S. Mittal, O. Tuzel, and P. Meer, "Semi-supervised kernel mean shift clustering", *IEEE Transactions on Pattern Analysis and Machine* Intelligence, vol. 36, no. 6, pp. 1201-1215, 2014.

[21] T. Xu, I. Cheng, and M. Mandal. "Automated cavity detection of infectious pulmonary tuberculosis in chest radiographs," in Proc. IEEE EMBC'11, 2011, pp.5178-5181.

[22] S. Zhang, F. He, Y. Zhang, J. Wang, X. Mei and T. Feng. "Geometric active contour based approach for segmentation of high-resolution spaceborne SAR images. " *Journal of Systems Engineering and Electronics* 26, no. 1， pp. 69-76, 2015.

[23] S. Ghadimi, H. A. Moghaddam, R. Grebe, F. Wallois, "Skull Segmentation and Reconstruction From Newborn CT Images Using Coupled Level Sets."

IEEE Journal of Biomedical and Health Informatics,vol.20, no.2, pp.563 - 573, 2016.

[24] T. Liu, *et al*. "Medical image segmentation based on a hybrid region-based active contour model." *Computational and mathematical methods in medicine*, vol. 2014, 10 pages, 2014. vol.2014, 2014.

[25] K. Zhang, *et al*. "Active contours with selective local or global segmentation: a new formulation and level set method. " *Image and Vision computing,* vol. 28, no.4, pp. 668-676, 2010.

[26] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris and R. L. Buckner, "Open Access Series of Imaging Studies (OASIS):cross-sectional MRI data in young, middle aged, nondemented, and demented older adults", Journal of Cognitive Neuroscience, no. 19, pp. 1498-1507, 2007.

[27] B. A. Ardekani, A. H. Bachman, K. Figarsky, and J. J. Sidtis, "Corpus callosum shape changes in early Alzheimer's disease: an MRI study using the OASIS brain database", *Brain Structure and Function*, vol. 219, no. 1, pp. 343-352, 2013.

[28] T. Xu, M. Mandal, R. Long, I. Cheng, and A. Basu, "An edge-region force guided active shape approach for automatic lung field detection in chest radiographs", Computerized Medical Imaging and Graphics, vol. 36, no.6, pp.452-63, Sep. 2012.

Appendix A

Matlab Code

The implementation and evaluation of the algorithm developed in this thesis has been performed using Matlab R2011a on a computer with Intel Core i7-3667U CPU 2GHz and 8GB RAM. In this appendix, I provide with some information about the MATLAB code. The list of MATLAB functions of the proposed system is shown in Table A.1.

In the beginning, a user can start the segmentation by running CCSeg.m which will initialize the graphical user interface. The GUI provides options for different tasks, and it will call different Matlab functions, as the steps move forward.

Table A.1. List of MATLAB Functions

No.	Function	Task	Page
1	CCSeg.m	Graphical User Interface	55
2	CC_AMSCluster	AMS Clustering	60
3	AMS_iter	AMS iteration	64
3	CC_Initial_Contour_Generator	CC Initial Contour Generation	66
4	template_matching	Calculates matching score images between a template and an image	69
5	CC_Final_Seg	Calculates a final CC boundary	72

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function varargout = CCSeg(varargin)
% CCSeg is a function that will initialize the graphical user
% interface of the proposed system.
% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @CCSeg_OpeningFcn, ...
                   'gui_OutputFcn',  @CCSeg_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
```

```
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
end


% --- Executes just before CCSeg is made visible.
function CCSeg_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to CCSeg (see VARARGIN)
set(handles.Btn_CCSeg,'visible','off');
% Choose default command line output for CCSeg
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
end


% --- Outputs from this function are returned to the command line.
function varargout = CCSeg_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Get default command line output from handles structure
varargout{1} = handles.output;
end


% --- Executes on button press in Btn_LoadIm.
function Btn_LoadIm_Callback(hObject, eventdata, handles)
% hObject    handle to Btn_LoadIm (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Clear Previous Result
set(handles.Btn_CCSeg,'visible','on');

% Load Image
% global inputimage;
[sFileName,sPath]= uigetfile('*.jpg;*.gif','Load Image');
%Get Directory & Image Info.
filename = [sPath,sFileName];
% ImgInfo = imfinfo(filename);
inputimage = imread(filename);
handles.inputimage = inputimage;
% Update handles structure
guidata(hObject, handles);
% Show the original image
handles.himg1 = imshow(inputimage,'Parent',handles.ImageWindow);
end

% --- Executes on button press in Btn_GenerateIC.
```

```
function Btn_GenerateIC_Callback(hObject, eventdata, handles)
% hObject    handle to Btn_GenerateIC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

IM = handles.inputimage;

% If input RGB image, convert it to gray-scale image
if ndims(IM) == 3
    IM = rgb2gray(IM);
end

% Generate CC Cluster
[CC_Cluster,Cluster_Image] = CC_AMSCluster(IM);

% Load template image
TP = rgb2gray(imread('ROI.jpg'));
% Generate initial contour
[CC_Initial_Contour, mask] =
CC_Initial_Contour_Generater(IM,TP,CC_Cluster);

% display the boudary with yellow contour in the original brain
image
% Show the original image
handles.himg1 = imshow(IM,'Parent',handles.ImageWindow);
hold on;
% Plot the contour
[A,B] = find(CC_Initial_Contour);
row = A(1); col = B(1);
```

```
contour = bwtraceboundary(CC_Initial_Contour, [row, col], 'W', 8,
length(A), 'counterclockwise');

plot(contour(:,2),contour(:,1),'y','LineWidth',3);
hold off;

handles.mask = mask;
% Update handles structure
guidata(hObject, handles);
end

% --- Executes on button press in Btn_CCSeg.
function Btn_CCSeg_Callback(hObject, eventdata, handles)
% hObject    handle to Btn_CCSeg (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
IM = handles.inputimage;

%% CC Seg Algorithm start here
if ndims(IM) == 3
    IM = rgb2gray(IM);
end

mask = handles.mask;

% % Generate CC Cluster
[Result] = CC_Final_Seg(IM,mask);

% Show the Original Image
handles.himg1 = imshow(IM,'Parent',handles.ImageWindow);
```

```
hold on;

% Display final result
h1 = contour(Result,[0 0],'y','linewidth',3);
hold on;
h2 = contour(Result,[0 0],'y','linewidth',3);

hold off;
end

% --- Executes on key press with focus on Btn_LoadIm and none of its
controls.
function Btn_LoadIm_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to Btn_LoadIm (see GCBO)
% eventdata  structure with the following fields (see UICONTROL)
%   Key: name of the key that was pressed, in lower case
%   Character: character interpretation of the key(s) that was
pressed
%   Modifier: name(s) of the modifier key(s) (i.e., control, shift)
pressed
% handles    structure with handles and user data (see GUIDATA)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [CC_Cluster,Cluster_Image] = CC_AMSCluster(IM)
% CC_AMSClusternew is a function that uses the adaptive mean shift
% (AMS) algorithm to cluster the input brain MRI based on grayscale
% intensity.
%
```

```
% Input:
%   IM - input brain MRI
%
% Output:
%   CC_Cluster - Cluster containing the CC region
%   Cluster_Image - generated cluster map
%
% Copyright (c) Dr. Yakov Keselman
% Department of Computer Science
% Rutgers University, Piscataway
% http:/www.yashma.org/yakovkeselman
%
% Changed by Yue Li, Univeristy of Alberta.

IM = double(IM);
[N,M] = size(IM);
Image_Min_Size = 0.005*N*M;

% Compute the total number of pixels in the brain MRI
Total = numel(IM);

% Selection of K (the number of neighbors that is
% considered in calculating the adaptive bandwidth)
numK = round(0.1*N*M);

% Establish a feature space - grayscale intensity only
gray_feature = zeros(1, max(IM(:))+1);
pro_gray_feature = zeros(1, max(IM(:))+1);
for i = 1:N
   for j = 1:M
```

```
      gray_feature(IM(i, j)+1) = gray_feature(IM(i, j)+1)+1;
   end
end

% Compute probability density function
pro_gray_feature = gray_feature./Total;

% Plot Grayscale Histogram
bar(pro_gray_feature);
set(gca,'XLim',[0 255]);
xlabel('Range of grayscale intensities');
ylabel('Probability of grayscale intensities');

% Initialization
ams_gray = gray_feature;
k = 0; L = length(ams_gray);

ColIm = reshape(IM,N*M,1);

while sum(ams_gray) > Image_Min_Size
   Non_zero_pos = find(ams_gray > 0);
   if isempty(Non_zero_pos) == 1
      break;
   end

   Initial = Non_zero_pos(1);

   [Mode, Number_values, Mode_bandwidth] = AMS_iter (ams_gray,
Initial,ColIm,numK);
   Mode = round(Mode);
```

```
    k = k+1;
    Modes(k) = Mode;
    Mode_bandwidths(k) = Mode_bandwidth;
    Initial_Value(k) = Non_zero_pos(1);
    if Mode-Mode_bandwidth > 1
        ams_gray(min(Initial, Mode-Mode_bandwidth): ...
        min(L, Mode+Mode_bandwidth)) = ...
        zeros(size(ams_gray(min(Initial, Mode-Mode_bandwidth): ...
        min(L, Mode+Mode_bandwidth))));
       else
           ams_gray(1:min(L, Mode+Mode_bandwidth)) = ...
               zeros(size(ams_gray(1:min(L, Mode+Mode_bandwidth))));
       end
    end

Cluster_Image = zeros(N,M);
Temp_Temp = zeros(N,M);
Temp_Cluster_Image = zeros(N,M,k);
B(1) = 0; C(1) = 0;
for i = 1:k
    To_group = find((IM >= max(1, Modes(i) - Mode_bandwidths(i)))...
        & (IM < min(L, Modes(i) + Mode_bandwidths(i))));
    Cluster_Image(To_group) = i;

% IM == 0 added to provide for background detection
To_zero = find(IM == 0);
Cluster_Image(To_zero) = 0;

Segmented = Cluster_Image;
Segments = size(Modes, 2);
```

```
Slices = zeros(size(IM));


CC_Cluster = zeros(size(IM));
count = 0;
percentage = zeros(1,Segments);
   for i = 1:Segments
      Slice = (Segmented == i);
      % some image processing - taken from morphology file
      Slice = bwmorph(Slice, 'majority');
      Slice = (Slice == 0); % negation
      Slice = bwmorph(Slice, 'majority');
      Slice = (Slice == 0); % negation
      percentage = sum(Slice(:))/Total;
      percentages(i) = percentage;
      if (i >= Cut) && (percentage >= 0.1)
          count = count + 1;
          AA = ((Slice(:,:)>0)*255);
          BB = imerode(AA, strel('rectangle', [3 3]));
          CC_Cluster(:,:,count) = BB;
       end
   end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Mode, NumberPixel,Mode_bandwidth] = AMS_iter (ams_gray,
Initial,ColIm,numK)

% AMS_iter is a function that iterates adaptive mean shift steps.
%
% Input:
```

```
%   ams_gray - Probability density function of grayscale intensity
%   Initial - An initial point
%   ColIm - Columnized image grayscale intensities
%   numK - The number of neighbors that is considered in calculating
%          the adaptive bandwidth for each data point
%
% Output:
%   Mode - Generated mode
%   NumberPixel - The number of pixels for each cluster
%   Mode_bandwidth - The bandwidth for generated mode
%
% Copyright (c) Dr. Romeil Sandhu
% Department of Electrical and Computer Engineering
% Georgia Institute of Technology
%
% Changed by Yue Li, Univeristy of Alberta.

NextPoint = round(Initial);
CurrentPoint = 0;
L = size(ams_gray, 2);

    diffBetweenIterations = 10;
    TAMS = 1;
    while diffBetweenIterations >= TAMS
        CurrentPoint = NextPoint;
        pointDist = abs(CurrentPoint - ColIm);

        [sortedDist,indexInOrig] = sort(pointDist);
        Distance = sortedDist(numk);
        bandwidth = round(Distance);
```

```
        Range = [max(1,CurrentPoint - bandwidth):...
            min(L, CurrentPoint + bandwidth)];
        Window = ams_gray(Range);
        numerator = full(Window * Range');
        NumberPixel = full(sum(Window));
        if numerator > 0
            NextPoint = round(numerator/NumberPixel);
        end
        diffBetweenIterations = abs(CurrentPoint - NextPoint);
end
Mode = NextPoint;
Mode_bandwidth = bandwidth;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [CC_Initial_Contour,mask] = CC_Initial_Contour_Generator
(IM,TP,CC_Cluster)
% CC_Initial_Contour_Generator is a function that will automatically
% generate an initial contour using the proposed AIC technique.
%
% Input:
%   IM - Input gray-scale brain MRI
%   TP - Input CC template Image
%   CC_Cluster - Input CC Cluster
%
% Output:
%   CC_Initial_Contour - Output Generated CC Initial Contour
%   mask - region inside the generated CC initial contour
```

```
% Read image and template
ori_im = double(IM);
tem = double(TP);

[rowT,colT] = size(tem);
row = size(CC_Cluster,1);
col = size(CC_Cluster,2);
im = zeros(row,col);

num = size(CC_Cluster,3);
count = 1;

up = ceil(0.25*row);
bottom = ceil(0.5*row);
left = ceil(0.25*col);
right = ceil(0.75*col);

for i = 1 : num
    im = CC_Cluster(:,:,i);

    partim = im(up:bottom,left:right);

    % Calculate SSD and NCC between Template and Image
    [I_SSD,I_NCC] = template_matching(tem,partim);

    % Find maximum correspondence in I_NCC image
    [xa,ya] = find(I_NCC == max(I_NCC(:)));

    x = up + xa;
    y = left +ya;
```

```
    xx = x - ceil(rowT/2);
    yy = y - ceil(colT/2);

    % Extract the NCC region
    NCC_region = zeros(size(tem));
    NCC_region = im(xx:xx + rowT, yy:yy + colT);

    % Convert NCC_region into binary image
    bg = (NCC_region<=100);
    NCC_region(bg(:)) = 0;
    NCC_region(~bg(:)) = 255;
    NCC_region_bw = im2bw(NCC_region);

    % Extract the largest connected area in the NCC region
    [L, num] = bwlabel(NCC_region_bw);
    maxarea = 0;
    maxindex = 0;
    for i = 1:num
        temp = length(find(L==i));
        if (temp>maxarea)
        maxarea = temp;
        maxindex = i;
       end
    end
    bw = (L == maxindex);

    im = imerode(bw,strel('rectangle',[3 3]));

% Extract the boundary of largestest connected area in NCC region
```

```
    bw1 = logical(zeros(row,col));
    bw1(xx:xx+rowT,yy:yy+colT) = bw;
    bw2 = bwperim(bw1,8);
    CC_Initial_Contour = bw2;
    mask = bw1;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [I_NCC] = template_matching(T,I)
% template_matching is a function that calculates similarity score
% images between a template and an image. The similarity is
% calculated based on the normalized cross correlation (NCC)(see
% Eq.(3.10) on page 27)
%
% Inputs:
%   T - Grayscale template image
%   I - Grayscale input image
% Outputs:
%   I_NCC - The similarity score image, and the calculation of
%           similarity is based on normalized cross correlation
%
% Copyright (c) Dr. Dirk-Jan Kroon
% University of Twente

% Convert images to double
T = double(T);
I = double(I);
[I_NCC,Idata] = template_matching_gray(T,I);
```

```
function [I_NCC] = template_matching_gray(T,I)
% Calculate correlation output size  = input size + padding template
T_size = size(T); I_size = size(I);
outsize = I_size + T_size-1;

% calculate correlation in frequency domain
if(length(T_size) == 2)
    FT = fft2(rot90(T,2),outsize(1),outsize(2));
    if(isempty(IdataIn))
        Idata.FI = fft2(I,outsize(1),outsize(2));
    else
        Idata.FI = IdataIn.FI;
    end
    Icorr = real(ifft2(Idata.FI.* FT));
else
    FT = fftn(rot90_3D(T),outsize);
    FI = fftn(I,outsize);
    Icorr = real(ifftn(FI.* FT));
end

% Calculate Local Quadratic sum of Image and Template
if(isempty(IdataIn))
    Idata.LocalQSumI = local_sum(I.*I,T_size);
else
    Idata.LocalQSumI = IdataIn.LocalQSumI;
end

QSumT = sum(T(:).^2);
```

```
if (nargout>1)
    % Normalized cross correlation STD
    if(isempty(IdataIn))
        Idata.LocalSumI = local_sum(I,T_size);
    else
        Idata.LocalSumI = IdataIn.LocalSumI;
    end

    % Standard deviation
    if(isempty(IdataIn))
        Idata.stdI = sqrt(max(Idata.LocalQSumI- ...
(Idata.LocalSumI.^2)/numel(T),0) );
    else
        Idata.stdI = IdataIn.stdI;
    end
    stdT = sqrt(numel(T)-1)*std(T(:));
    % Mean compensation
    meanIT = Idata.LocalSumI*sum(T(:))/numel(T);
    I_NCC = 0.5+(Icorr-meanIT)./ (2*stdT*max(Idata.stdI,stdT/1e5));

    % Remove padding
    I_NCC = unpadarray(I_NCC,size(I));
end

function T = rot90_3D(T)
T = flipdim(flipdim(flipdim(T,1),2),3);

function B = unpadarray(A,Bsize)
Bstart = ceil((size(A)-Bsize)/2)+1;
Bend = Bstart+Bsize-1;
```

```
if(ndims(A)==2)
    B = A(Bstart(1):Bend(1),Bstart(2):Bend(2));
elseif(ndims(A)==3)
    B = A(Bstart(1):Bend(1),Bstart(2):Bend(2),Bstart(3):Bend(3));
end

function local_sum_I= local_sum(I,T_size)
% Add padding to the image
B = padarray(I,T_size);

% Calculate for each pixel the sum of the region around it,
% with the region the size of the template.
    s = cumsum(B,1);
    c = s(1+T_size(1):end-1,:)-s(1:end-T_size(1)-1,:);
    s = cumsum(c,2);
    local_sum_I = s(:,1+T_size(2):end-1)-s(:,1:end-T_size(2)-1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [result,pin,pout] = CC_Final_Seg(IM,mask)

% CC_Final_Seg is a function that uses the Geometric Active Contour
% (GAC)algorithm to generate the final CC contour on the input
% brain MRI.
%
% Input:
%
% IM - input grayscale brain MR image
% mask - Region inside the generated initial CC contour
%
```

```
% output:
% result - Final CC contour
% pin - Pdf of pixel intensities in interior region of the
% generated CC contour
% pout - Pdf of pixel intensities in exterior region of the
% generated CC contour
%
% Copyright (c) Dr. Romeil Sandhu
% Department of Electrical and Computer Engineering
% Georgia Institute of Technology
% http://iss.bu.edu/tannenba/people/Romeil.Sandhu/Romeil.Sandhu.html
%
% Changed by Yue Li, Univeristy of Alberta.

% Load an image
IM = double(IM);

% Compute sdf function
phi = bwdist(mask) - bwdist(1-mask)+im2double(mask);

% Run active contour
[result,pin,pout] = Tryphon_NB(IM, phi);
end



function [phi,pin,pout] = Tryphon_NB(IM, phi)
% Get image size dimensions
[dimR, dimC] = size(IM);

dt = 0.4;
```

```
alpha = 0.005;
flag_approx = 1;

% Set intensity range
h = 1:1:256;

% Run active contour
for s = 1:150

    % Find appropriate narrowband
    index = find(phi < 2 & phi > -2);

    % Index for rows/cols
    [nrow, ncol] = ind2sub(size(phi), index);

    % Initialization
    K = zeros(size(index));
    d_T = zeros(size(index));

    % Set inside/outside curve points
    in_pt = find(phi < 0) ;
    out_pt = find(phi > 0) ;

    % Compute probability density function
    pin = create_pdf(IM(in_pt))';
    pout = create_pdf(IM(out_pt))';

    % Area computation
    Ain = numel(in_pt);
    Aout = numel(out_pt);
```

```
    % Compute "global" energy terms
    T = log((pin + eps)./(pout + eps));
    D = sqrt(mean(T.^2) - mean(T).^2);

    % Cycle through narrowband points
    for i = 1:numel(index)

        % Index properly
        nr = nrow(i);  nc = ncol(i); ind = index(i);

        % Boundary conditions
        if((nr+1) >= nrow) nr = row-1; end
        if((nr-1) <= 0)    nr = 2;    end
        if((nc+1) >= ncol) nc = col-1; end
        if((nc-1) <= 0)    nc = 2;    end

        % Derivatives for kappa
        phi_x =  phi(nr, nc+1) - phi(nr, nc-1);
        phi_y =  phi(nr+1, nc) - phi(nr-1, nc);
        phi_xx = phi(nr, nc+1) - 2*phi(nr, nc) + phi(nr,nc-1);
        phi_yy = phi(nr+1, nc) - 2*phi(nr, nc) + phi(nr-1,nc);
        phi_xy = -0.25*phi(nr-1,nc-1)- ...
0.25*phi(nr+1,nc+1)+0.25*phi(nr-1,nc+1)+0.25*phi(nr+1,nc-1);

        % Curvature gradient flow
        norm = sqrt(phi_x.^2 + phi_y.^2);
        K(i) = ((phi_x.^2.*phi_yy + phi_y.^2.*phi_xx - ...
2*phi_x.*phi_y.*phi_xy)./...
                (phi_x.^2 + phi_y.^2 +eps).^(3/2)).*norm;
```

```
        % Energy to Minimize
        delta  = Dirac2(h - (IM(ind)+1), 1);
        delta  = delta/(sum(delta));

        % Compute curve energy
        G = (1/Ain - 1/Aout) - delta.*(1./(Ain*pin+eps) + ...
1./(Aout*pout+eps));
        d_T(i) = (1./D) * (mean(T.*G)-mean(T)*mean(G))*norm;
        if(flag_approx); d_T(i) = .25*sign(d_T(i)); end;
    end

    % Combine energy terms
    if(flag_approx); alpha = .125; end;
    e = (d_T)+alpha*K;

    % Find max energy
    max_e = max(abs(e));

    % Compute update of level set function
    phi(index) = phi(index) + (0.4/(max_e+eps))*(e);

    % Redistance every 10 iterations
    if(mod(s,10) == 0); phi = sussman(phi, .5); end

% Display evolution contour
h1 = contour(phi,[0 0],'y','linewidth',3);
hold on;
h2 = contour(phi,[0 0],'y','linewidth',3);
hold off;
```

```
    end
end
```