Chapter 2

ROI Boundary Extraction

2.1 Introduction

In this chapter, an algorithm is presented for the extraction of the region of interest (ROI) enclosed by the outer boundary of the imaged object. Object boundary extraction from a motion affected image can be difficult, especially if the motion function of the object is unknown. Although there are many algorithms presently available for solving segmentation tasks, they can be easily misled by the ghost artifacts and blurring in the background of the image. Therefore, a two step background clearing algorithm is proposed in this chapter, in order to facilitate object boundary extraction. The first step involves selection of the least motion-affected views. Two possible methods are discussed for this purpose, and their capabilities and limitations are examined. The first method involves computing the X-directional inverse Fourier transform (IFT) widths of each view in the k-space, whereas the second method involves an entropy minimization criterion [81] for the suppression of motion induced blur. The second step in the image background clearing algorithm involves cancellation of the remaining ghost artifacts,

using a fuzzy model. This model is formulated to represent the image background region. Both the steps involved in background clearing tend to increase the number of dark pixels in the image. The contour extraction is performed using an improved active contour model (snakes) [109], which can overcome several limitations of the conventional snake model, while preserving all the desirable properties associated with it.

Popular post-processing methods for motion artifact correction, use the projection onto convex sets (POCS) algorithm [67] to perform motion estimation and correction [59][65][76][77][85]. The success of POCS largely depends on the convex sets, defined by the constraints imposed via available *a priori* knowledge. Finite support constraint [67] is one of the most popular convex sets employed in the past. However, no convincing methods have been proposed as to how the finite support, marked by the outer boundary of the region of interest (ROI), is extracted from the motion affected image. Furthermore, the ROI boundary is also extensively used in motion parameter estimation [61][78][84]. Minimization of pixel energy outside the ROI has been used not only to acquire useful motion information but also for eventual motion artifact suppression in MR images [53][54][61][62][69][74][76][77]. Therefore, it is imperative to extract the ROI boundary with high accuracy, for post-processing methods to be successful.

There are two methods of ROI boundary extraction directly from motion affected images, proposed in the past. Zoroofi et. al. [78] propose that simple thresholding would produce an estimate for the ROI. This method seems impractical for two reasons. Firstly, the resulting ROI entirely depends on the thresholding value which is chosen arbitrarily. Secondly, in the presence of strong ghost artifacts outside ROI, thresholding will not be able to eliminate these high intensity ghosts without seriously compromising the edges of the original object. The result will be an erroneous estimate for the boundary of ROI. Hedley et. al. [69] propose a method of manually fitting a boundary by an expert. This technique may be

successful in some cases where the corrupted image warrant edge estimation by a trained user, in spite of the presence of artifacts. However, such a method produces unacceptable results in the midst of severe rotational motion, where the spurious ghost edges conceal the real boundary of the object. Manual curve fitting also demand considerable time and concentration on the part of the expert, rendering this method inefficient.

This chapter describes a robust, efficient, reliable and semi-automatic algorithm for the purpose of ROI contour extraction from the motion affected MR images. Since the nature of the motion artifacts differ according to the motion itself, deblurring techniques found in the past [110]-[113] are of limited use in suppressing motion induced blur in the image background. Most deblurring methods assume knowledge on the blur operator or the original image [110]-[111]. However, in blind image restoration [112]-[113], blur operator is considered unknown and is posed as a function (i.e. point spread function) to be identified either initially or as an integral part of the restoration procedure. In this application, the primary objective is to clear the image background from motion induced blurring rather than to enhance the image features or extract the blur operator. Therefore, it makes little sense to pursue blind image restoration techniques due to the high algorithmic complexity and the computational load involved.

2.2 View Selection

In T_1 weighted spin echo MR imaging [114], the time spent on acquiring a spin echo signal is relatively short compared to the time interval between two acquisitions. This is usually obvious from the choice of *echo time* (TE) [114] and the *repetition time* (T_R) [114], where $T_R \gg TE$. Therefore, the object can be assumed to be stationary during the acquisition of a single view (i.e. a phase encode). However, the object may move during the time interval between two

consecutive phase encodes (i.e. inter-view motion). This inter-view motion is a usual occurrence in rigid motion [62].

In order to formulate the image-background clearing algorithm, we observe the following properties of the acquired MRI signal space (i.e. k-space):

- Each view provides, in part, the spatial frequency information of the object in a particular location and orientation. For rigid in-plane motion, the imaged object is constant over the scanned views. However, its location and orientation may differ from view to view.
- If N views of data are acquired, ranging from $V = 0$ to $V = N - 1$, the views in the middle of the k-space (i.e. near $V = \frac{N}{2}$) contain the intensity information, whereas the views towards the ends of the k-space (i.e. near $V = 0$ and $V = N - 1$) contain the edge information.
- The effect of including $V = \frac{N}{2} \pm i$ (where $i = 1, 2, ..., \frac{N}{2} - 1$) view information in the reconstruction stage is twofold. Firstly, it improves stationary edge information, but secondly it also introduces motion induced blur and ghosting artifacts.

It should be noted that this is a simplified description of the k-space. However, it is sufficient for the formulation of the algorithm for background clearing. According to the above properties of k-space, it is possible to form an image of the object undergoing rigid in-plane motion, using only the low frequency information given by the views in the vicinity of view $V = \frac{N}{2}$. The resulting image exhibits the orientation and location of the object at view $V = \frac{N}{2}$, provided that the views concerned are not subjected to severe motion. Therefore, it is important to select views that match the orientation and location of the object to that of view $V = \frac{N}{2}$. However, a problem arises as motion information is unavailable

for each view. Therefore, two possible methods of view selection are proposed, to successfully overcome this potential problem.

2.2.1 X-directional Inverse Fourier Transform Method

This method is based on a low-frequency image reconstructed using only a small number of views with comparable X-directional inverse Fourier transform (IFT) widths. For convenience, this method will be referred to as Method 1. Each view of data is subjected to inverse Fourier transform (IFT) separately, to establish the resulting X-directional IFT width associated with each view. The width of this X-directional IFT is shown to provide the projection width of the imaged object on to the x-axis [61]. Therefore, two open *snake* contours are used on either side of the X-directional IFT to extract this projection width. A detailed description of the snake model is given in Section 2.4 of this chapter.

The X-directional IFT of the ${k_m}^{th}$ view data is given by

$$ {S'}_X(x, k_m) = \frac{1}{N} \sum_{k_x=-\frac{N}{2}}^{\frac{N}{2}-1} S'(k_x, k_m) e^{i(\frac{2\pi}{N})xk_x} \tag{2.1} $$

where $S'(k_x, k_m)$ represent the acquired k-space data for the ${k_m}^{th}$ view, and N is the number of samples in a view. If the object $m(x, y)$ at its orientation at the ${k_m}^{th}$ view is denoted by $\hat{m}(x, y, k_m)$, $S'(k_x, k_m)$ is given by the following equation:

$$ S'(k_x, k_m) = \sum_{x=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{y=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{m}(x, y, k_m) e^{-i(\frac{2\pi}{N})(xk_x + yk_m)} \tag{2.2} $$

Performing the inverse Fourier transform along the X-direction, the following equation for ${S'}_X(x, k_m)$ is obtained:

$$ {S'}_X(x, k_m) = \frac{1}{N} \sum_{y=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{m}(x, y, k_m) e^{-i(\frac{2\pi}{N})yk_m} \tag{2.3} $$

As shown in Figure 2.1, if the object $\hat{m}(x, y, k_m)$ is contained within x_{min} and x_{max} at the acquisition of the ${k_m}^{th}$ view, the Y-directional projection width w_l is given by

$$w_l = x_{max} - x_{min} + 1 \tag{2.4}$$

Furthermore, from Equation 2.3, ${S'}_X(x, k_m) = 0$ for $x < x_{min}$ and $x > x_{max}$. Therefore, the width of the X-directional IFT equates to the Y-directional projection width w_l of $\hat{m}(x, y, k_m)$. As shown in Figure 2.1, a particular value of w_l corresponds to a unique orientation of the scanned object provided that the object is unsymmetric. For a symmetric object, it is possible to have multiple orientations matching the same w_l value. The number of such possible orientations depends on the number of possible axes of symmetry.

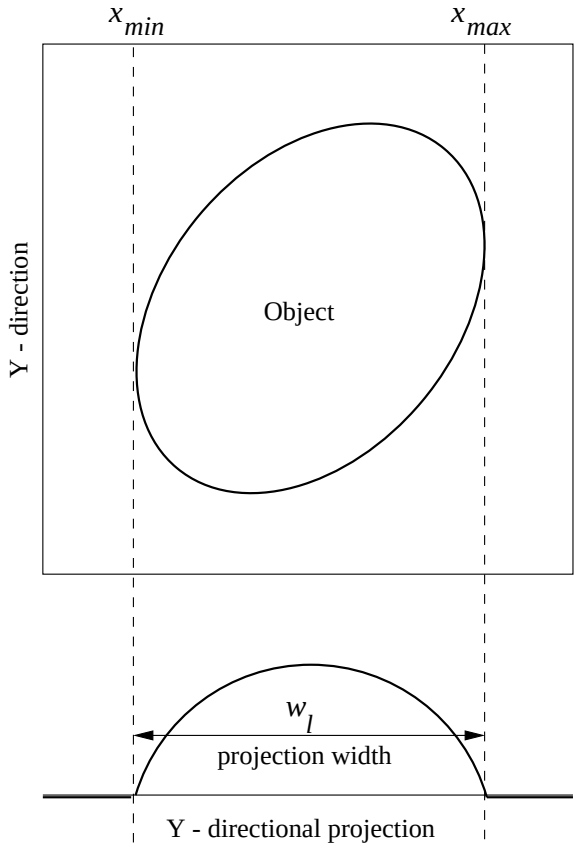

Figure 2.1: Illustration of Y-directional projection width w_l.

The set of selected views $\mathcal{V}$, based on w_l is hence given by

$$\mathcal{V} = \left\{ v(i) \quad | \quad |w_l(i) - w_l\left(\frac{N}{2}\right)| < \epsilon \right\} \tag{2.5}$$

where $i = 0, 1, ..., N-1$ and ϵ is a user defined tolerance limit.

2.2.2 Entropy Minimization Method

This method is based on minimising the entropy of the image pixel intensities. For convenience, this method will be referred to as Method 2.

Since the bulk of the image energy is located in the low frequencies, the view $V = \frac{N}{2}$ is associated with the maximum energy in most practical scenarios. Therefore, the view $V = \frac{N}{2}$ is chosen as the motion free reference view which describes the object at a fixed location and orientation. When adding the views $V = \frac{N}{2} \pm i$ for $i = 1, 2, ..., l$ (where $l < \frac{N}{2}$), the following four options are available:

- Option 1: discard the data in both $V = \frac{N}{2} - i$ and $V = \frac{N}{2} + i$ views by replacing each data point with 0.
- Option 2: include the data in both the views $V = \frac{N}{2} - i$ and $V = \frac{N}{2} + i$ without any change.
- Option 3: include the data in view $V = \frac{N}{2} - i$, while replacing the data in $V = \frac{N}{2} + i$ by the complex conjugate values of $V = \frac{N}{2} - i$ as given in the following equation:

$$S(k_x, k_y) = S^*(N - k_x, N - k_y) \tag{2.6}$$

  where $k_y = \frac{N}{2} + i$ and $k_x = 0, 1, ..., N-1$. The Hermetian symmetry implied by this equation may not hold in general, for the acquired MR data under clinical settings. However, if there are significant artifacts in the image background, caused by the absence of Hermitian symmetry, option 1 or

option 2 will be selected instead. It should be noted that at this stage, the primary objective is to reduce the intensity of motion artifacts in the image background rather than to correct the motion corrupted data.

- Option 4: include the data in view $V = \frac{N}{2} + i$, while replacing the data in $V = \frac{N}{2} - i$ by the complex conjugate values of $V = \frac{N}{2} + i$ as given in the Equation 2.6 with $k_y = \frac{N}{2} - i$.

It should be noted that Option 1 can suppress motion artifacts associated with views $V = \frac{N}{2} \pm i$. However, it can also introduce truncation artifacts, due to the narrow spectral window available. Truncation artifacts in the form of Gibbs ringing in the spatial domain, can blur the image background. Therefore, the selection of options can be viewed as balancing the effects of motion artifacts and Gibbs ringing. The selection is made so that the chosen option minimizes the intensity of the motion artifacts in the background by increasing the number of dark pixels. Hence, it is important to define a cost function which can indicate the intensity of artifacts in the image background. The entropy of the image is used as the required cost function, since it favours the alteration to the data that tends to increase the number of dark pixels. The entropy of the image is given by:

$$E = \sum_{j=1}^{N^2} -r_j \ln r_j \tag{2.7}$$

where S is the number of image pixels and r_j is the normalized intensity given by

$$r_j = \frac{I_j}{I_{tot}} \tag{2.8}$$

where I_j is the modulus of the complex value of the j^{th} image pixel intensity and I_{tot} is given by

$$I_{tot} = \sqrt{\sum_{j=1}^{N^2} {I_j}^2} \tag{2.9}$$

From the Equation 2.7, it is possible to define the entropy e_j at each pixel as

$$e_j = r_j \ln\left(\frac{1}{r_j}\right) \tag{2.10}$$

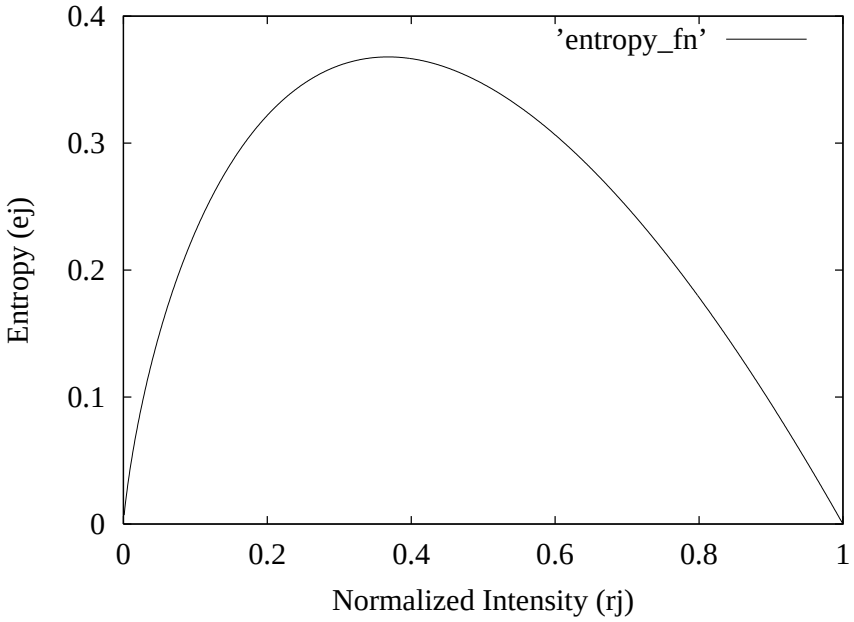

Figure 2.2: Entropy function e_j plotted against normalized intensity r_j.

In order to analyze the entropy at pixel level, e_j is plotted against normalized pixel intensity r_j in Figure 2.2. According to Figure 2.2, e_j is zero when $r_j = 0$ (i.e. $I_j = 0$) or 1 (i.e. $I_j = I_{tot}$). Therefore, e_j is minimum if a pixel is black or if all the image energy is located in a single pixel (i.e. all the remaining pixels are black). Hence, minimizing e_j clearly tends to increase the number of dark pixels. However, in most practical applications, the imaged object occupies more than just a few pixels, resulting in a large I_{tot} value compared to the maximum possible "intensity" at a single pixel (i.e. $I_{tot} \gg \max\{I_j \quad \text{for} \quad j = 1, 2, ..., N^2\}$). For example, the Shepp & Logan phantom [121] has $I_{tot} = 10572$ and maximum $I_j = 255$. Therefore, in practice, $r_j \ll 1$ (typically $\max\{r_j\} \approx 10^{-2}$). The view selection criterion based on the image entropy is hence given by

$$\text{Option}_{sel}(i) = \min_{Option_k} \left\{ \sum_{j=1}^{N^2} r_j \ln\left(\frac{1}{r_j}\right) \right\} \tag{2.11}$$

where $i = 1, 2, ..., l$ and $k = 1, ..., 4$. It should be noted that the proposed criterion is only capable of choosing the best option from the given four options. Therefore, if all four options produce significant artifacts in the reconstructed image background, the algorithm will choose the option that creates least artifacts, although this option may still produce visible artifacts. Therefore, further processing is required after selecting the least motion affected views, in order to obtain sufficient suppression of motion artifacts in the image background.

2.3 Fuzzy Model

Each pixel value in the background can be affected by motion artifacts, Gibbs ringing or a combination of both these artifacts. Gibbs ringing tend to dominate when there are few views available for the image reconstruction. When additional view data are included, motion artifacts begin to dominate over the Gibbs ringing. Therefore, l images are reconstructed using views $[\frac{N}{2} - i, ..., \frac{N}{2}, ..., \frac{N}{2} + i]$ for $i = 1, 2, ..., l$. According to the intensity at each pixel, a fuzzy set [115] is defined, representing the image background. The membership function is given by

$$\mu_i(I_j) = \frac{1}{1 + \left(\frac{I_j}{\psi}\right)^2} \tag{2.12}$$

where constant ψ is set appropriately. As explained in Chapter 1 Section 1.8, fuzzy models are generated arbitrarily, based on the application. Therefore, the fuzzy parameter value ψ is also subjective rather than precise. After experimentation with the test data sets, it was concluded that a ψ value within the range [10, 50] give acceptable results. In the experiments described in this chapter, ψ value was fixed at 30 for all cases studied, to illustrate that the performance of the fuzzy model is not highly dependent on the selection of the ψ value.

According to equation (2.12), higher membership values are allocated for darker

pixels. This membership function represents a fuzzy or soft threshold. Therefore, a change in the value of ψ does not inflict significant change in the final result, compared to hard thresholding.

It is possible to construct l fuzzy sets using the l reconstructed images. The overall membership of any pixel location is computed using the fuzzy union [115] of the available l fuzzy sets as shown below:

$$\mu = \bigvee_{i=1}^{l} \mu_i \tag{2.13}$$

The "maximum" operator is used for the fuzzy union. Other s-norms [115] such as drastic sum, bounded sum, Einstein sum, algebraic sum or Hamacher sum could have been used for this purpose. However, the "maximum" operator is chosen since it produces the smallest value (i.e. the most conservative) out of the s-norms mentioned above [115]. Therefore, the value of the fuzzy union clearly favors the dark pixels, since it maximizes the membership of a pixel to be in the background. Therefore, the fuzzy union captures the instance where a particular background pixel location has the least effect from the motion artifacts and Gibbs ringing. This greatly facilitates successful clearing of the image background.

The resulting membership value is then mapped onto each image pixel, as shown below:

$$I_j = I_{max}(1 - \mu) \tag{2.14}$$

where I_j is the resulting intensity at the j^{th} pixel, and I_{max} is the maximum possible intensity. For gray scale images, I_{max} is set to 255.

It is interesting to note that the Method 2 for view selection described in the previous section, and the fuzzy model described in this section may prove useful in various signal processing applications where the signals are acquired in sequential blocks. The available *options* related to entropy minimization will be intrinsic to any particular application. Therefore, as described in Chapter 1

Section 1.8, the fuzzy model can also be formulated in many different forms to suit the application, by incorporating more variables such as ψ.

2.4 Review of Improved Active Contour Model

The overall goal of this chapter is to find a smooth contour that describes the boundary of the imaged object most accurately. Hence, after the background clearing stage, an improved snake model [109] is applied for the purpose of contour extraction. The active contour, also known as snake, has played an important role in medical image analysis [116], since it was first proposed by Kass [117] in 1987. It has proven to be effective in segmenting structures from an image by exploiting constraints derived from the image data together with *a priori* knowledge about the location, size and the shape of these structures. Traditional low-level image processing techniques (eg. thresholding, edge detection and region growing) consider only local information and are prone to making incorrect assumptions during its boundary element integration process resulting in erroneous object boundaries. The advantage of snakes is the integration of image segmentation and representation phases into a single step [116] which makes the subsequent analysis and interpretation of the segmented object more convenient. Furthermore, it is capable of accommodating the often significant variability of biological structures over time and across different individuals.

2.4.1 Snake Model

Much research work on the development and application of snakes has been reported in recent literature [109] [118]-[119]. However, the improved model [109] used in this thesis can not only overcome several limitations [118] of the original model [117], but can also preserve all the desirable properties associated

with it. This model is less sensitive to initial conditions due to the feedback mechanism [109]. However, it does not limit the shape recovery scheme of the model. The improved model is capable of extracting the desired contour without any over-run. One of the interesting aspects of this model is that it can extract concave and convex contours most accurately while at the same time able to bear the subjective contour by linking relevant broken edges. Figure 2.3 shows the flow diagram of the improved snake energy minimization process [109], where the snake energy function (E_{snake}) for a contour containing N distinct control points is given by

$$E_{snake} = \sum_{i}^{N} \{E_{int}(V_i) + P_{image}(V_i) - E_{feedback}(V_i)\} \tag{2.15}$$

where V_i is the i^{th} distinct control point and $E_{int}(V_i)$ represents the internal energy term which imposes the regularity on the curve by bending and stretching. This is given by

$$E_{int}(V_i) = \alpha(i)E_{cont}(V_i) + \beta(i)E_{curv}(V_i) \tag{2.16}$$

where the primes $\alpha(i)$ and $\beta(i)$ control the mechanical properties (ie. elasticity and rigidity) of the model. The E_{cont} and E_{curv} terms are given by

$$E_{cont}(V_i) = \frac{d - |V_i - V_{i-1}|}{\max_j\{d - \left|V_{i_j} - V_{i-1}\right|\}} \tag{2.17}$$

and

$$E_{curv}(V_i) = \frac{|V_{i-1} - 2V_i + V_{i+1}|^2}{\max_j\{\left|V_{i-1} - 2V_{i_j} + V_{i+1}\right|^2\}} \tag{2.18}$$

where V_{i_j} are the eight neighbourhood points of the control point V_i on the curve for $j = 0, 1, ..., 7$, and d is the average curve length.

$P_{image}(V_i)$ term in Equation 2.15 represents the image potential energy, which attracts the snake toward salient image attributes, such as lines, edges and texture. For edges, $P_{image}(V_i)$ is defined as follows:

$$P_{image}(V_i) = \gamma(i)P_{field-norm}(V_i) \tag{2.19}$$

where the prime $\gamma(i)$ is the weight for image attraction and $P_{field-norm}(V_i)$ is given by

$$P_{field-norm}(V_i) = \frac{P_{field}(V_i) - P_{field-min}}{P_{field-max} - P_{field-min}} \tag{2.20}$$

The term $P_{field}(V_i)$ represents the image gradient, whereas $P_{field-max}$ and $P_{field-min}$ are respectively the maximum and minimum values of $P_{field}(V_i)$ within the eight neighbourhood points of V_i.

The feedback energy term $E_{feedback}(V_i)$ in Equation 2.15 is given by

$$E_{feedback}(V_i) = f_{db}(i)\nabla P_{field}(V_i)\vec{n}(i) \tag{2.21}$$

where the prime $f_{db}(i)$ controls the feedback pulling or pushing on the snake. $\vec{n}(i)$ is the vector normal to the contour at the i^{th} control point. The term $\nabla P_{field}(V_i)$ is given by

$$\nabla P_{field}(V_i) = \begin{cases} \frac{P_{max} - \frac{\sum_{j=1}^{8} P_{field}(V_{i_j})}{8}}{P_{max}} & \text{if } P_{field}(V_i) < P_{level} \\ 0 & \text{if } P_{field}(V_i) \geq P_{level} \end{cases} \tag{2.22}$$

where P_{max} and P_{level} are respectively the maximum and threshold levels in the potential field of image features (i.e. edges in this particular application).

Using proper values for the parameters $\alpha(i)$, $\beta(i)$, $\gamma(i)$, $f_{db}(i)$ and introducing another parameter GAP, it is possible to control the behavior of the snake model in order to obtain a smooth contour which fits the object contour most accurately. The parameter GAP is the threshold of the average length between two adjacent points of the snake. A larger GAP value is useful in mitigating the formation of loops. However, a small GAP value may prompt the snake to capture the details of the objective contour.

According to the flow diagram in Figure 2.3, the parameter $f_{db}(i)$ controls the feedback strength of the model. For the cases in this study, where the final contour is obtained by continually shrinking the initial contour, a larger $f_{db}(i)$ propels the snake points along the normal direction to the curve, overcoming

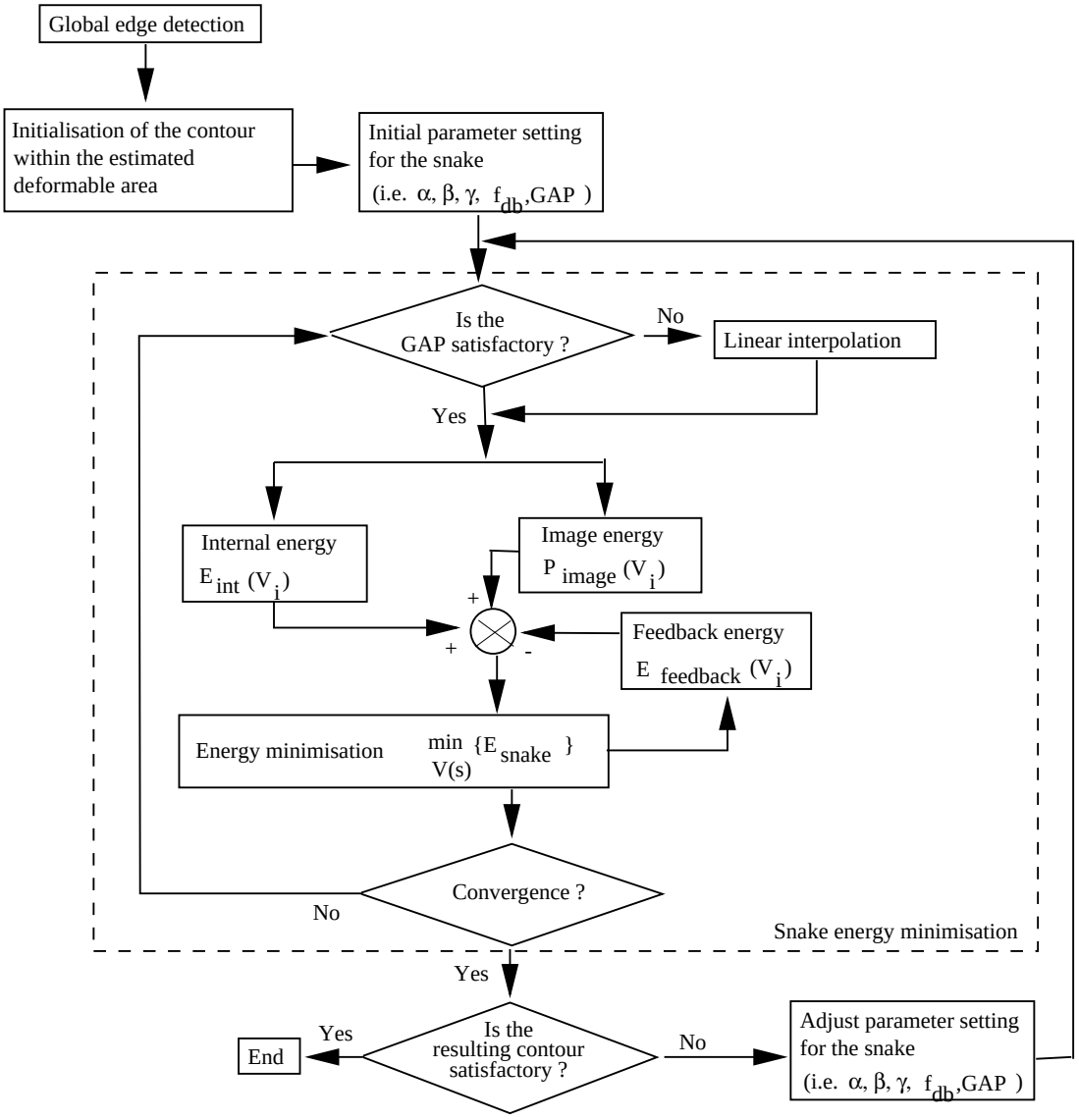

Figure 2.3: Flow chart for the improved snake energy minimization process.

its internal bending and stretching resistance in order to flow into concave or convex contour shapes. However, small $f_{db}(i)$ prompts the snake points to be more influenced by the bending energy to overcome spurious edge trappings or to bear subjective contours.

2.4.2 Graphical User Interface

The contour extraction algorithm based on snakes, described in the Section 2.4.1 has been implemented with a graphical user interface (GUI), in order to support interactive user inputs. The initial contours are suggested by the program, however, these contours can be altered using the keyboard or the mouse. When a particular image is presented, the program executes a function which analyzes 4×4 blocks from the periphery of the image to identify whether the pixel sum of these blocks are equal to zero. Based on this analysis, the program suggests an initial contour which passes through only these zero blocks. The program also supports pre-defined initial contours that can be used as default. In this thesis, the initial contour suggested by the program for the sagittal head slice (see Chapter 1 Section 1.9) experiment was used as the default initial contour for the other experiments. It was observed that analyzing 4×4 blocks consumes longer time compared to setting a default contour and adjusting it manually if needed. The sagittal head slice experiments were used for this initialization since the imaged object is largest in this case, hence the initial contour is guaranteed to be outside the imaged objects for other cases involving axial head slice and Shepp & Logan phantom [121]. If the object is larger than the imaged window, *open* snake initials are used. The extreme points of the open contours usually reside at the image window boundary, and are fixed during the snake energy minimization process. Examples of the use of closed and open snake contours can be found in simulation experiments in this chapter and Chapter 4.

The initial snake parameters are set by default to $\alpha = 0.5$, $\beta = 0.8$, $\gamma = 0.9$, $f_{db} = 0.4$ and $GAP = 5$. These parameters can be changed by the user, using the "parameter settings" dialog box. For a given initial contour and a parameter setting, the program displays the result of the converged snake when 60% of the image energy is within the converged contour. The user has the option of accepting the contour as the final result, continue snake energy minimization,

or re-start the process using a different set of parameters. This user interactive step is shown in Figure 2.3 with the decision step "Is the resulting contour satisfactory?". The GUI also provide graphical setting of *bending*, *stretching*, *attraction* and *feedback* using 4 slider bars in the parameter settings dialog box, which increase or decrease the values of α, β, γ and f_{db}.

Therefore, the snake parameter setting is performed manually, and the initial snake contour is specified automatically, semi-automatically or manually, using the GUI.

2.5 Simulation Results

Simulation studies are performed using the Shepp & Logan phantom [121], by subjecting the phantom to step-wise and continuous rotations.

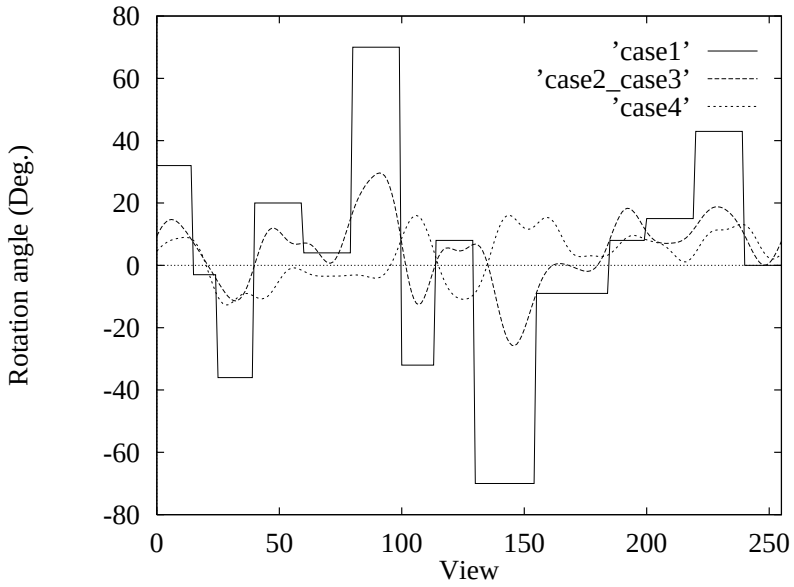

Figure 2.4: Rotational motion parameters for each phase encode for the four cases studied.

The rotational motion was simulated by rotating the phantom by an angle θ prior to obtaining the spatial frequency information for each view. The corrupted

$N(=256)$ samples of data for the k_y^{th} view are given by

$$S(k_x, k_y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} m(\bar{x}, \bar{y}) e^{[-i\frac{2\pi}{N}(xk_x + yk_y)]} \tag{2.23}$$

where $\bar{x} = x\cos\theta - y\sin\theta$ and $\bar{y} = x\sin\theta + y\cos\theta$. k_x is the sample number in the frequency encoded direction, $m(x, y)$ is the original Shepp & Logan phantom and θ is the angle of rotation for the $k_y{}^{th}$ view. The proposed method is also

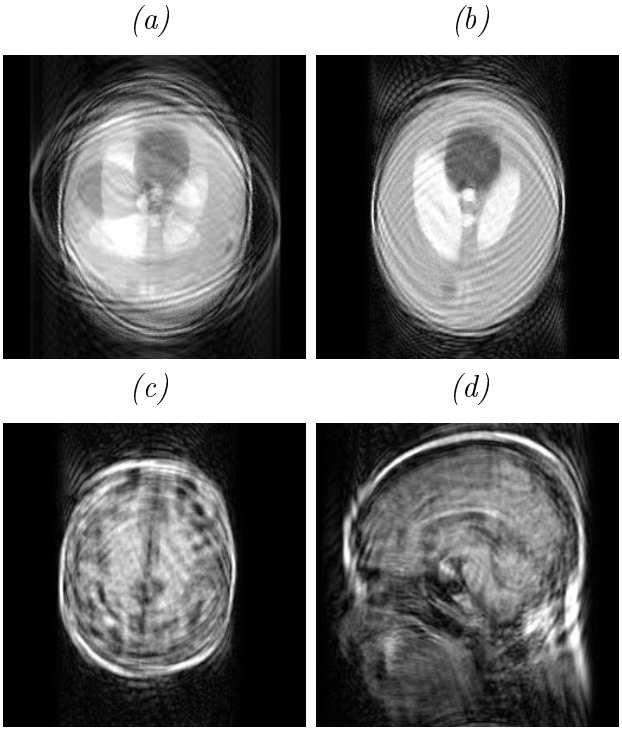

Figure 2.5: Images reconstructed using 2D IFFT method, from MR data affected by in-plane rotational motion. The maximum angular span (MAS) is given for each case: (a) case 1 - S&L phantom, $MAS = \pm 70^o$; (b) case 2 - S&L phantom, $MAS = \pm 30^o$; (c) case 3 - axial head slice - $MAS = \pm 30^o$; (d) case 4 - sagittal head slice - $MAS = \pm 15^o$.

examined with spin echo MR images subjected to in-plane rotational motion. Figure 2.4 shows the view numbers and corresponding angles of rotation for four different cases studied. The first two cases involve the Shepp & Logan phantom [121], whereas cases 3 and 4 involve axial and sagittal MR images of a human head. The maximum angular span is $\pm 70^o$ for case 1, $\pm 30^o$ for cases 2 and 3, and $\pm 15^o$ for case 4. The reconstructed images for these four cases, using

conventional 2D IFFT method [5], are shown in Figure 2.5. The total number of acquired views is 256, and the resultant image size is 256×256 pixels.

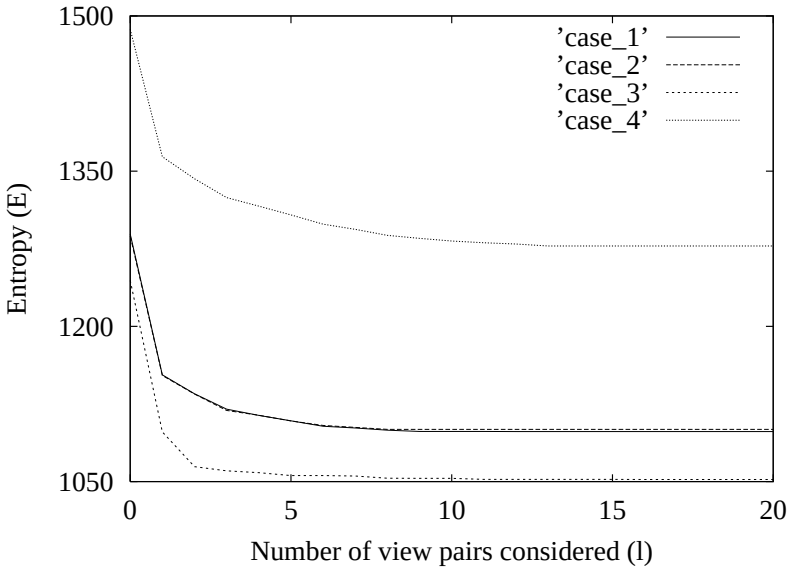

Figure 2.6: Entropy minimization as a function of considered number of view pairs l.

Least motion affected views are selected using the two methods described in Section 2.2, in order to determine the effectiveness of each method in clearing the image-background. In the case of Method 1, the tolerance value (ϵ) is set to 2. For the Method 2, the value of l is set to 15, which is considered to be a sufficient number of views to extract the outer boundary of the imaged object. l value could have also been set to its maximum ($\frac{N}{2} - 1$). However, this causes unnecessary computations, since most of the views where $l > 15$ are discarded subjected to Option 1. Figure 2.6 shows the rate of entropy minimization as a function of l for the four cases considered in this study. Since the total energy embedded in each view decreases with increasing l, even if few more views are selected subjected to Options 2 - 4 for $l > 15$, the effect of such inclusion is minimum with respect to the entropy of the image, suppression of truncation artifacts and background clearing. However, l can be increased up to ($\frac{N}{2} - 1$) if needed, at the expense of computational efficiency. Therefore, in this study $l = 15$ is chosen to construct an acceptable low-frequency image while saving valuable computational time. It

is expected that any disadvantage of not selecting $l = (\frac{N}{2} - 1)$ is adequately offset by the application of the fuzzy model and the resulting computational efficiency.

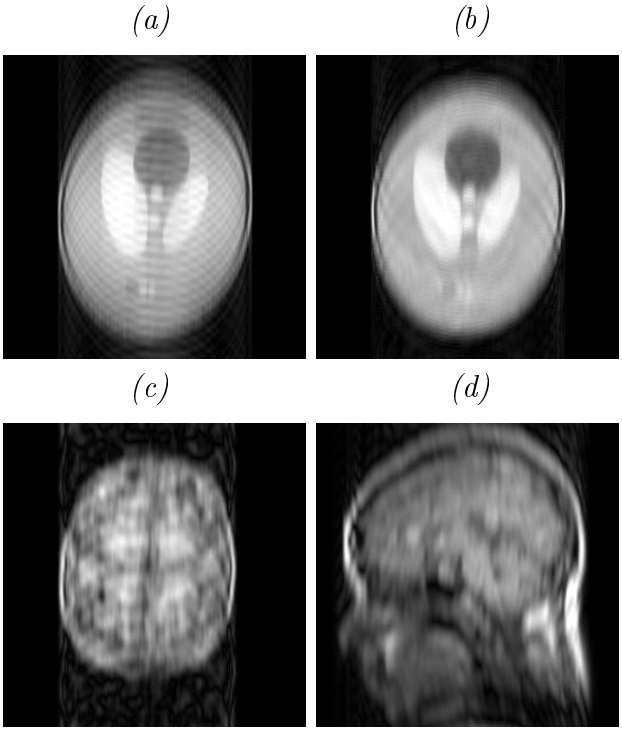

Figure 2.7: Images reconstructed using the selected views of X-directional IFFT method, using maximum of ($l = 15$) view pairs. The pixel energy (e_p) in the image background is given for each case: (a) case 1 ($e_p = 63.751$) ; (b) case 2 ($e_p = 48.608$) ; (c) case 3 ($e_p = 197.341$) ; (d) case 4 ($e_p = 268.733$) .

The images reconstructed using the selected views of Method 1 and Method 2 are shown in Figures 2.7 and 2.8 respectively. It can be seen that there are visible artifacts remaining in the image background, especially for cases 3 and 4, which are subjected to continuous rotations. A comparison of the pixel energy in the image background is given in Table 2.1, for low-frequency images constructed by the selected views using the Method 1 and the Method 2. The superior performance of the Method 2 can be attributed to the avoidance of selecting views corresponding to multiple orientations with same X-directional IFT width (w_l). For symmetric objects, such as those used in Cases 1 - 3, the Method 2 shows a significant improvement over the Method 1.

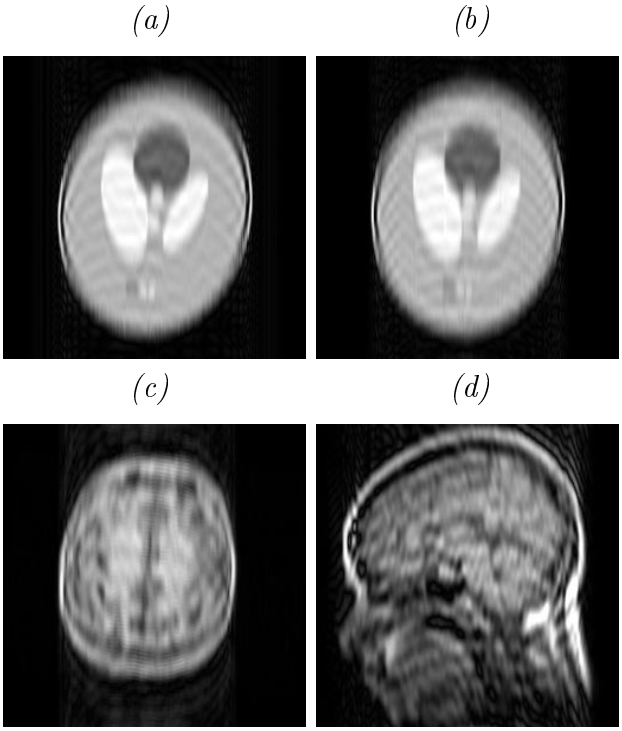

Figure 2.8: Images reconstructed using the selected views of Entropy minimization method, using maximum of ($l = 15$) view pairs. The pixel energy (e_p) in the image background is given for each case: (a) case 1 ($e_p = 46.088$) ; (b) case 2 ($e_p = 40.891$) ; (c) case 3 ($e_p = 186.463$) ; (d) case 4 ($e_p = 263.891$).

Case number	Recons. image using 2D-IFFT	Recons. image using Method 1	Recons. image using Method 2
1	905.982	63.751	46.088
2	417.099	48.608	40.891
3	242.696	197.341	186.463
4	586.502	268.733	263.891

Table 2.1: Comparison of pixel energy in the image background, for reconstructed images with motion artifacts shown in Figure 2.5, after the view selection using the Method 1 and the Method 2.

The pixel energy in the image background is given by

$$e_p = \frac{1}{p} \sum_{j \notin ROI}^{S} |I_j|^2 \tag{2.24}$$

where p is the total number of pixels in the image background. The ROI boundaries used in these calculations are the final results shown in Figure 2.11.

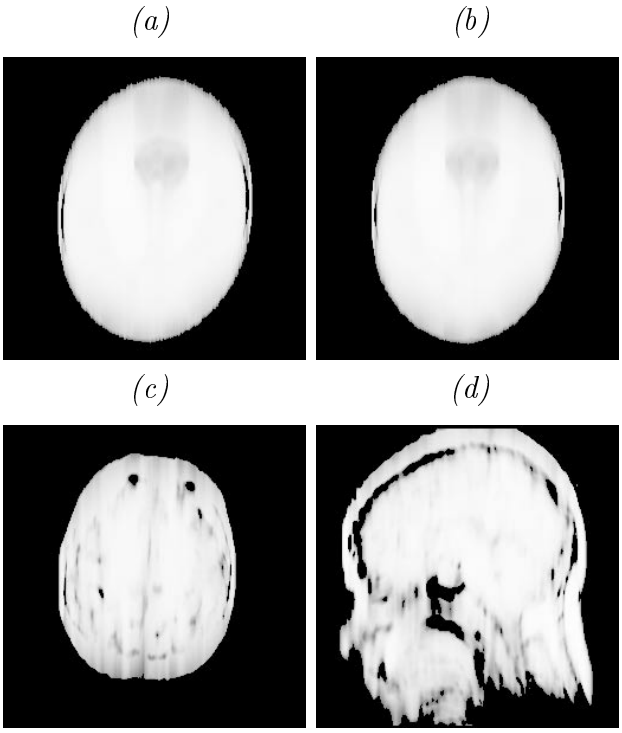

Figure 2.9: Images with cleared background, using the proposed fuzzy algorithm on images reconstructed via entropy minimization method. The value of the fuzzy model parameter ψ is set to 30 in all the studied cases. The pixel energy (e_p) in the image background is given for each case: (a) case 1 ($e_p = 3.029$) ; (b) case 2 ($e_p = 1.054$) ; (c) case 3 ($e_p = 0.060$) ; (d) case 4 ($e_p = 17.902$) .

In order to suppress the remaining artifacts in the background, the fuzzy model described in Section 2.3 is employed. The value of ψ is set to 30 in all the studied cases. Since the superior performance of the Method 2 for view selection is established, only the images constructed using Method 2 are subjected to further processing using the fuzzy model. The resulting images are shown in Figure 2.9. Notice that the image background is cleared in spite of the loss of object details. However, the goal is to clear the image background in order to facilitate the contour extraction, rather than preserve the details within the

object. Therefore, the proposed fuzzy model of the image background, performs successfully in achieving the desired objective. A comparison of the pixel energy in the image background is given in Table 2.2 for prior to and after applying the fuzzy model.

Case number	Before applying fuzzy algorithm	After applying fuzzy algorithm
1	46.088	3.029
2	40.891	1.054
3	186.463	0.060
4	263.891	17.902

Table 2.2: Comparison of pixel energy in the image background, before and after applying the proposed fuzzy algorithm.

The object boundary is extracted using the improved active contour model described in Section 2.4.1. The initial contour is placed using the GUI as described in Section 2.4.2. Since the snake model is less sensitive to the shape and position of the initial contour, it can be any closed initial guess in the image background (shown in Figure 2.10). The extracted ROI contours are displayed in Figure 2.11. The snake parameters [109] used for the convergence of the snake are also given for each case. Figure 2.12 compares the final ROI contours extracted using the Method 1 and the Method 2, for the Case 1, to illustrate the improved smoothness and accuracy achieved by the use of Method 2.

In order to examine the performance of the proposed ROI extraction scheme, the resulting contours are compared with the motion free images of the objects. It should be noted that in reality, motion free images are unavailable. However, in this study, the motion free images have been used for the sole purpose of performance evaluation and are not required by the proposed algorithm. The extracted ROI contours are overlaid on the motion free images in the best fitting orientation in Figure 2.13. Notice that the estimated contour does not invade the object in all the four different cases. Furthermore, the estimated boundary very

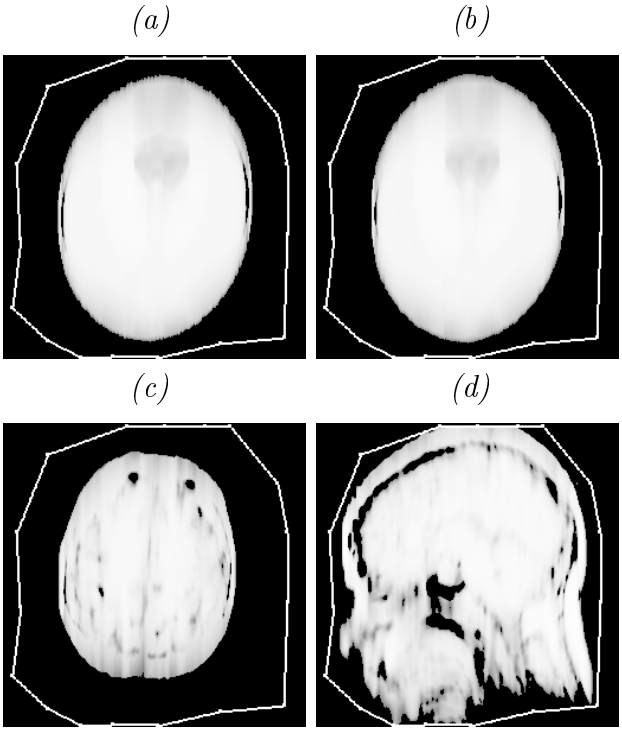

Figure 2.10: Initial contours for object boundary extraction, placed using the 4 block analysis of the image presented for case 4 (see Section 2.4.2 for details). Since the cases 1 to 3 also satisfy this initial contour, it is used as the default without alterations via GUI of snake energy minimization program [109].: (a) case 1 (using default); (b) case 2 (using default); (c) case 3 (using default); (d) case 4 (using 4 block analysis).

closely matches the object outer boundary. These are two important characteristics of the proposed method. Invasion of the object can lead to catastrophic consequences at both the motion correction and motion parameter estimation stages. Since both these stages heavily rely on minimizing the pixel energy outside the ROI [78] [53], an invasion of ROI boundary inside the object can remove parts of the object from the reconstructed image by forcing those pixel energies to zero. However, allowing significantly larger estimates for the boundary of ROI results in ambiguous estimation of motion parameters [62] and slow convergence of reconstruction algorithms that use POCS. Therefore, it is important to ensure that the ROI boundary is on or outside but adjacent to the outer boundary of the object.

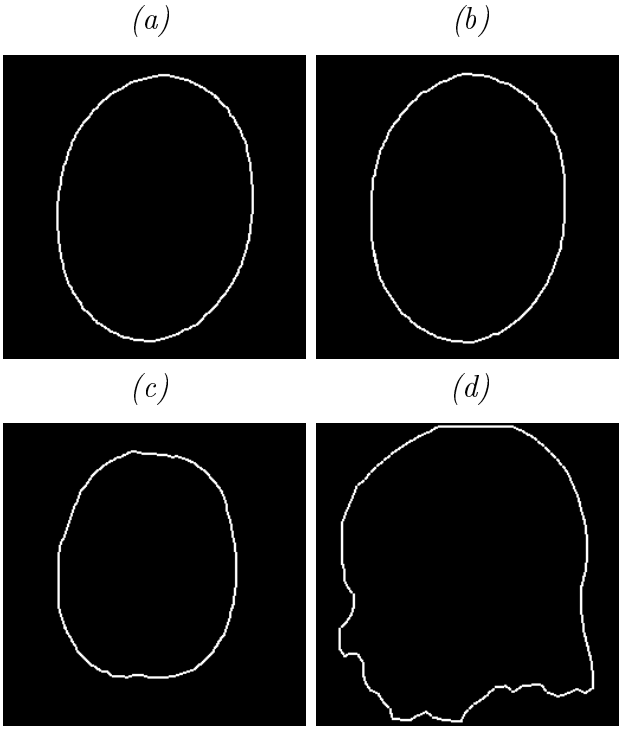

Figure 2.11: Extracted contours after convergence of the snake. The snake parameters [109] used for each of the studied cases are also given: (a) case 1 - ($\alpha = 0.5$, $\beta = 0.85$, $\gamma = 0.9$, $f_{db} = 0.4$, $GAP = 5$); (b) case 2 - ($\alpha = 0.5$, $\beta = 0.85$, $\gamma = 0.9$, $f_{db} = 0.4$, $GAP = 5$); (c) case 3 - ($\alpha = 0.5$, $\beta = 0.85$, $\gamma = 0.9$, $f_{db} = 0.4$, $GAP = 5$); (d) case 4 - ($\alpha = 0.54$, $\beta = 0.4$, $\gamma = 0.7$, $f_{db} = 0.54$, $GAP = 9$).

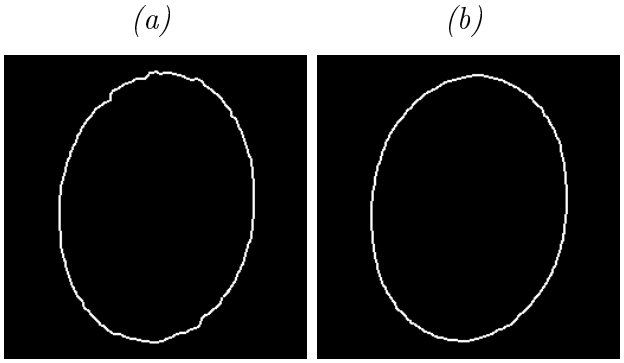

Figure 2.12: Comparison of the final ROI contours extracted using Method 1 and Method 2 as part of the background clearing algorithm, for Case 1: (a) Method 1; (b) Method 2.

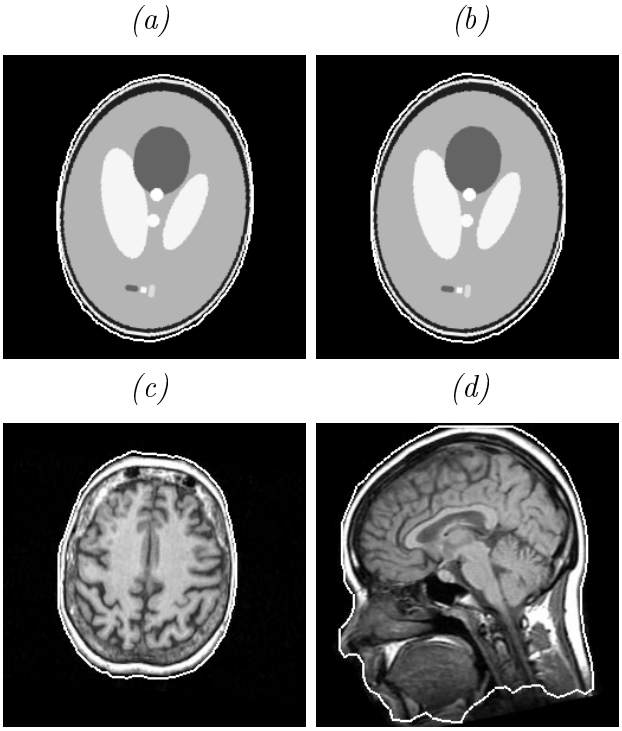

Figure 2.13: Extracted ROI contours overlaid on the motion free images in the best fitting orientation: (a) case 1; (b) case 2; (c) case 3; (d) case 4.

In order to compare the results with that of direct thresholding [78], images shown in Figure 2.14 are obtained. These images are produced directly from the artifact images shown in Figure 2.5 using twice the ψ value used in the fuzzy algorithm (see Section 2.3), as the threshold value in each case. The contour extraction algorithm described in Section 2.4 is used to identify the object boundary as shown in Figure 2.15. The initial contours and the snake parameters used are identical to those used with the proposed algorithm, except in Case 4, where the parameters are changed to prevent the formation of loops. However, the residual ghost edges resulted in an inferior ROI boundary (compared to that shown in Figure 2.13). The threshold values are then further increased until all the background artifacts are eliminated (see Figure 2.16). However, this results in the ROI boundary invading the object, due to the loss of edge details as shown in Figure 2.17. It is obvious from these results, that the proposed algorithm produces higher quality ROI boundaries compared to thresholding.

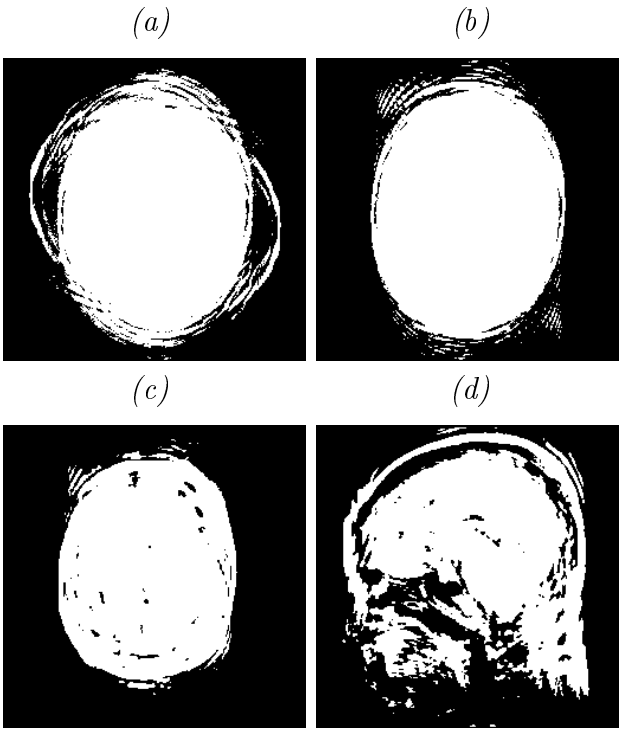

Figure 2.14: Images produced by direct thresholding the artifact images: (a) case 1 ($th = 60$); (b) case 2 ($th = 60$); (c) case 3 ($th = 60$); (d) case 4 ($th = 60$).

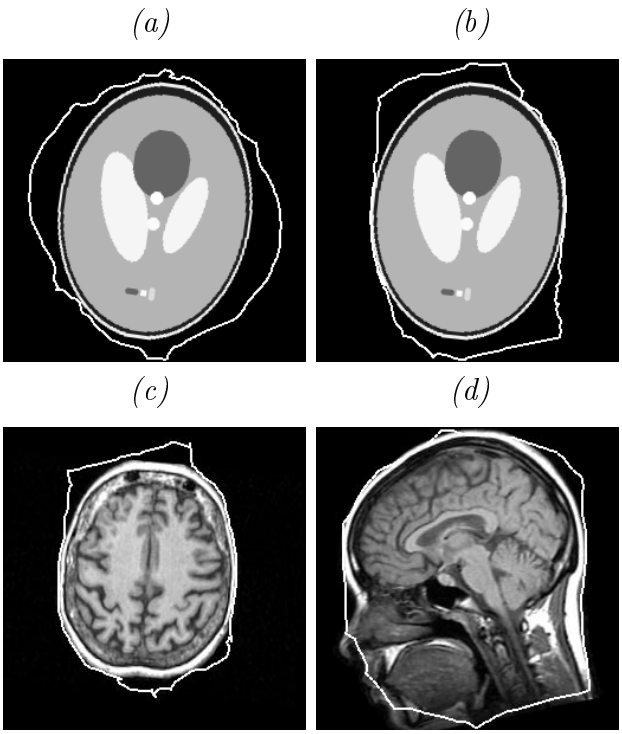

Figure 2.15: Extracted ROI contours from threshold images of Figure 2.14, overlaid on the motion free images: (a) case 1; (b) case 2; (c) case 3; (d) case 4.

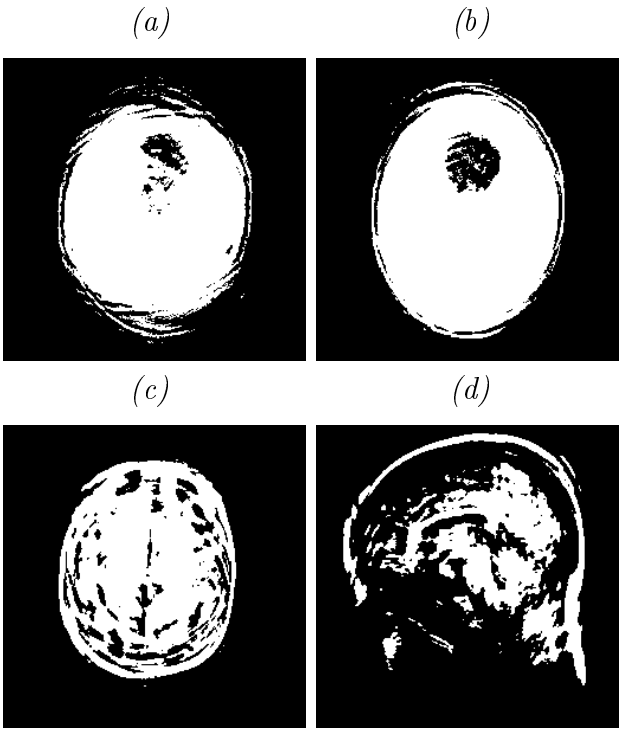

Figure 2.16: Images produced by increasing the threshold value: (a) case 1 ($th = 130$); (b) case 2 ($th = 110$); (c) case 3 ($th = 110$); (d) case 4 ($th = 130$).

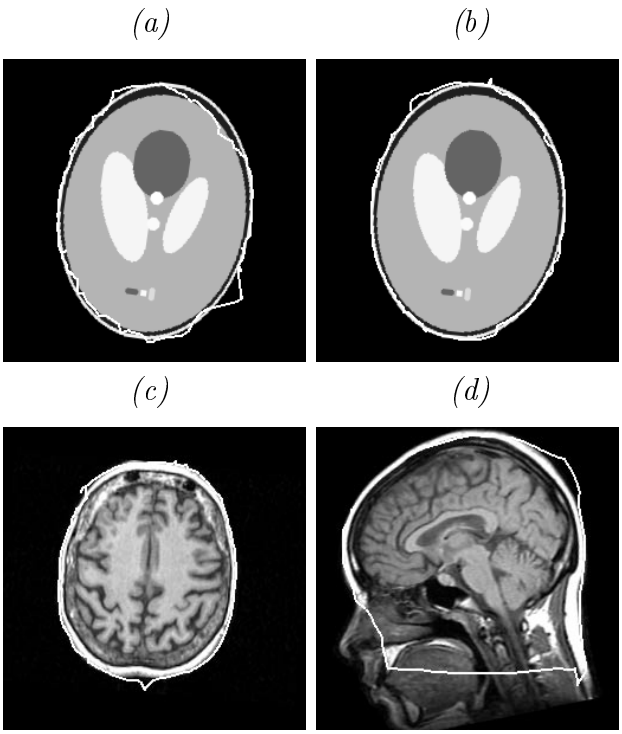

Figure 2.17: Extracted ROI contours from threshold images of Figure 2.16, overlaid on the motion free images: (a) case 1; (b) case 2; (c) case 3; (d) case 4.

The ROI boundary extraction times using the proposed method on a Sun SPARC 2 workstation (X Windows) varies from 2 minutes to 7.5 minutes depending on the input data. Most time is consumed at the view selection stage, and each snake convergence typically takes less than a minute, whereas the fuzzy model only takes a couple of seconds. Since the program involves a GUI, the execution time is also influenced by the graphics capability of the terminal and the number of times the initial snake parameters have to be adjusted before achieving a satisfactory solution.

2.6 Summary

The simulation studies involving Shepp & Logan phantom [121] and spin echo MR images indicate that the proposed algorithm is capable of extracting the ROI boundary from motion affected MR images, with high accuracy and reliability without invading the object area. It is also established that the view selection based on entropy minimization produces superior results compared to the method based on X-directional IFT width. The method incorporating entropy minimization is robust, less subjective and can be applied under a wide variety of motion models including severe rotations. It has the potential to be extended to handle many other types of motion such as expansion and out-of-plane rotations. Furthermore, it can be combined with intelligent motion parameter estimation schemes and POCS based motion correction algorithms to effectively suppress motion artifacts in MR images, as shown in the following chapters.