Chapter 4

Rotational Motion Estimation

4.1 Introduction

A major obstacle to the success of post-processing artifact correction techniques in MRI is the scarcity of reliable motion estimation algorithms. Most on-line motion estimation schemes demand (patient) preparation, modifications to standard spin-echo pulse sequences and increased scanning times. Therefore, off-line motion estimation algorithms have gained interest in this research area. However, the existing algorithms are plagued by high computational and time demands that restrict the estimation capability to only a few motion parameters. This chapter presents an efficient off-line motion estimation algorithm with applications to in-plane rotational motion artifact correction in MRI. The algorithm is based on maximizing the *similarity* among the k-space data subjected to angular overlap. The initial guesses are derived from measuring projection width of X-directional inverse Fourier transforms of the acquired k-space views.

One technique for obtaining motion information is to employ navigator echoes that are interleaved into the imaging sequence [42][58][125]. However, this

method can only provide information on translational motion along one axis. Rotational motion can be determined by tracking the location of externally attached markers [58] with one or more conventional navigator echoes. This method requires patient preparation. Use of orbital navigator echoes avoid this requirement [125]. However, it involves measurement of MR signals along circular k-space trajectories with extensive modifications to the standard spin echo sequences, and k-space re-gridding problems [79] [126] at the image reconstruction stage.

Motion estimation at the post-processing stage has also attracted wide attention. In previously proposed techniques, such as, the minimum energy method [78], entropy focus criterion [81] and projection onto convex sets (POCS) with finite support constraint [62], the underlying idea is to minimize the pixel energy outside the region of interest (ROI). However, the pixel energy at a particular point on the image is dependent on the magnitude and phase of all the k-space data, which prohibits this method from estimating the rotations corresponding to each view at a time. Therefore, all the unknown rotations have to be estimated at once using a single optimization strategy. For m unknowns, the optimization problem becomes m dimensional, with a prohibitively large search space for larger rotations, resulting in longer estimation times. One method for simplifying this problem is to use a specific rotational model, which can describe all unknowns by a single parameter, as described in Chapter 3 for the case of rotational motion at constant angular velocity. However, such simplification can jeopardize the generality of the method and performs poorly outside the specified model. This chapter presents a simplified in-plane rotational motion estimation scheme without the loss of generality, to improve the estimation efficiency and image reconstruction time. The proposed algorithm is also capable of providing a reliability measure for each estimated rotation angle to facilitate and guide the artifact correction algorithm.

4.2 Similarity Criterion

If the imaged object is rotated at an angle θ_r, at the acquisition of ${k_m}^{th}$ view, the relationship between the acquired corrupted signal S' and the required k-space data S is given by the Equation 3.5 of Chapter 3. Since the motion is considered to be an inter-view effect, θ_r is constant during the acquisition of all N samples in the frequency encoded k_x direction. If the ${k_m}^{th}$ view is rotated by an angle θ and the ${k_n}^{th}$ view is rotated by a different angle α, the acquired data samples denoted by $S'(k_{xm}, k_m)$ and $S'(k_{xn}, k_n)$ both represent the same spatial frequency information in the uncorrupted k-space, provided that the following conditions are satisfied:

$$k_{xm}\cos\theta + k_m\sin\theta = k_{xn}\cos\alpha + k_n\sin\alpha \quad (4.1)$$

$$\text{and} \quad -k_{xm}\sin\theta + k_m\cos\theta = -k_{xn}\sin\alpha + k_n\cos\alpha \quad (4.2)$$

where $k_m \neq k_n$, $\theta \neq \alpha$ and $0 \leq k_{xm}, k_{xn} \leq N$.

Therefore, when projected onto the uncorrupted k-space, data overlap will occur between the ${k_{xm}}^{th}$ sample of the ${k_m}^{th}$ view and the ${k_{xn}}^{th}$ sample of the ${k_n}^{th}$ view, as shown in Figure 4.1. Data overlapping can occur between any number of different views, rotated at different angles, hence producing significant regions of data overlap at large angle rotations.

The proposed angle estimation scheme uses the *similarity* among the k-space values acquired in different views, at data overlap regions, to find the corresponding rotation angle for each view. The reference view is always chosen to be the $\left(\frac{N}{2}\right)^{th}$ view, since it contains the bulk of the image energy. Therefore, the orientation of the object at the acquisition of the $\left(\frac{N}{2}\right)^{th}$ view is associated with the angle 0^o. The consequent angle estimations for the rest of the views in the k-space are derived relative to the $\left(\frac{N}{2}\right)^{th}$ view. The angle estimations are performed each view at a time, in an ordered sequence for view numbers $\frac{N}{2} \pm v$,

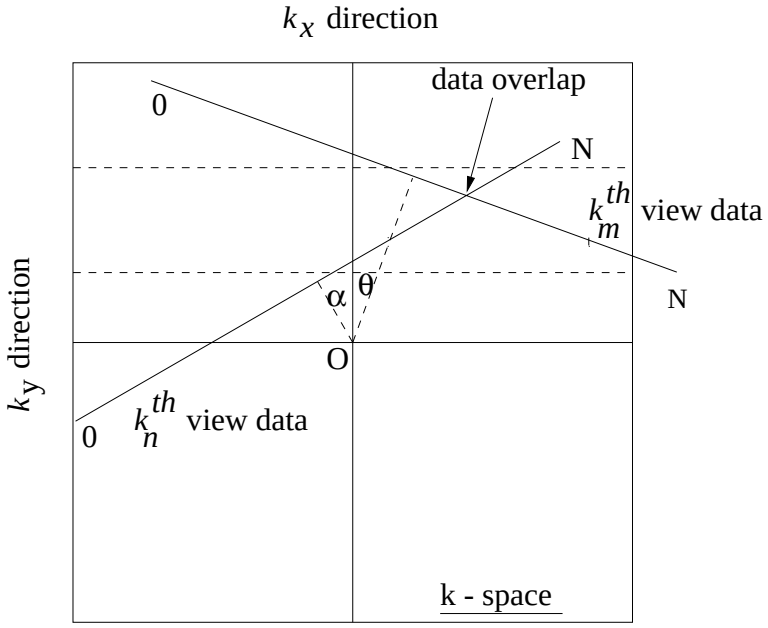


Figure 4.1: Illustration of data overlap between two views, on the k-space of the imaged object.

where $v = 1, 2, ..., \frac{N}{2} - 1$. The k-space data for each view is corrected using the bilinear superposition algorithm [78], prior to proceeding with the angle estimations for consequent views. This method effectively transforms an N-dimensional optimization problem into N one-dimensional optimizations, with a significant reduction in computations.

Considering the i^{th} overlap point in the k-space for the ${k_m}^{th}$ view (i.e. the intersection point of the ${k_m}^{th}$ view with the $\left(\frac{N}{2}\right)^{th}$ view or with another already corrected view), for a supposed rotation angle θ_j of the ${k_m}^{th}$ view, the similarity criterion is defined by

$$C_i(k_m, \theta_j) = 1.0 - \left[\frac{(|S'_m| - |S'_0|)^2}{|S'_m|^2 + |S'_0|^2}\right] \tag{4.3}$$

where the value S'_m belongs to the ${k_m}^{th}$ view, and S'_0 belongs to $\left(\frac{N}{2}\right)^{th}$ view or one of the already corrected views. Only the magnitude values of acquired k-space data are used, in order to avoid phase errors imposed by concurrent translational motion and magnetic field inhomogeneity. Therefore, the similarity criterion is valid for rotational motion in the imaging plane, with or without the presence of translational motion.

The similarity criterion is formulated so that it produces the value 1.0 if the overlapping k-space values are equal in magnitude (i.e. $|S'_m| = |S'_0|$). The value of $C_i(k_m, \theta_j)$ decreases for an increasing difference between $|S'_m|$ and $|S'_0|$. If they are *vastly different* (i.e. $(|S'_m| - |S'_0|)^2 \gg |S'_m|^2 + |S'_0|^2$), a large negative value is produced. In order to improve the estimation robustness against noisy k-space data, the $C_i(k_m, \theta_j)$ values are averaged over all the overlapping points associated with the view k_m. The average over p overlap points is given by

$$\bar{C}(k_m, \theta_j) = \frac{1}{p} \sum_{i=1}^{p} C_i(k_m, \theta_j) \tag{4.4}$$

Therefore, the optimization problem for angle estimation at the ${k_m}^{th}$ view is formulated by seeking θ_j that maximizes the average *similarity* at all overlap points, between the ${k_m}^{th}$ view and the already corrected views.

$$\theta_{est}(k_m) = \max_{\theta_j} \left[\bar{C}(k_m, \theta_j)\right] \tag{4.5}$$

The reliability of each estimate is expressed by a membership function [131], $\mu\left[\theta_{est}\left(k_m\right)\right]$, which maps the maximum average *similarity* at the ${k_m}^{th}$ view to the real interval $[0, 1]$. That is,

$$\mu\left[\theta_{est}\left(k_m\right)\right] = \begin{cases} q & \text{if } 0 \leq q \leq 1 \\ 0 & \text{if } q < 0 \\ 1 & \text{if } q > 1 \end{cases} \tag{4.6}$$

where

$$q = \frac{\bar{C}(k_m, \theta_{est}) + 1.0}{2.0} \tag{4.7}$$

For values $\left|\bar{C}(k_m, \theta_{est})\right| \leq 1.0$, the estimated rotation angles are included in a fuzzy set [131] with membership function $\mu\left[\theta_{est}\left(k_m\right)\right]$. This membership function is consequently used not only for interpolation of less reliable estimates but also for modification of the artifact correction algorithm.

4.3 Selection of Search Space

In order to minimize the search space of test angles, the search is limited to a specified interval near the initial guesses. These initial guesses are derived using an approach based on measuring the projection width of X-directional inverse Fourier transforms (IFT) of each view. A similar method was first proposed by Wood et. al. [127]. This method was given little attention in the past mainly due to its limited accuracy and dependence on the object shape. However, in this application, it is used only as a starting point for angle estimations.

The algorithm for deriving initial guesses, is shown in the block diagram of Figure 4.2, and is described as follows:

Firstly, 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 (see Chapter 2). Therefore, two open *snake* contours are used on either side of the X-directional IFT to extract this projection width. The outer boundary of the object is extracted by the algorithm described in Chapter 2, using a closed snake contour. The extracted boundary is then rotated by known step-wise angles, in order to establish the projected widths it casts on the x-axis. The results of this stage are used to create a look-up table, indicating the rotation angle associated with each projection width. Hence, the X-directional IFT width of each view can be converted to an associated rotation angle. If multiple angles of rotation cast the same projection width on the x-axis (i.e. due to symmetric objects), all these possible angles are clustered in the search space. The major steps involved in this algorithm are discussed in detail, in the following sections.

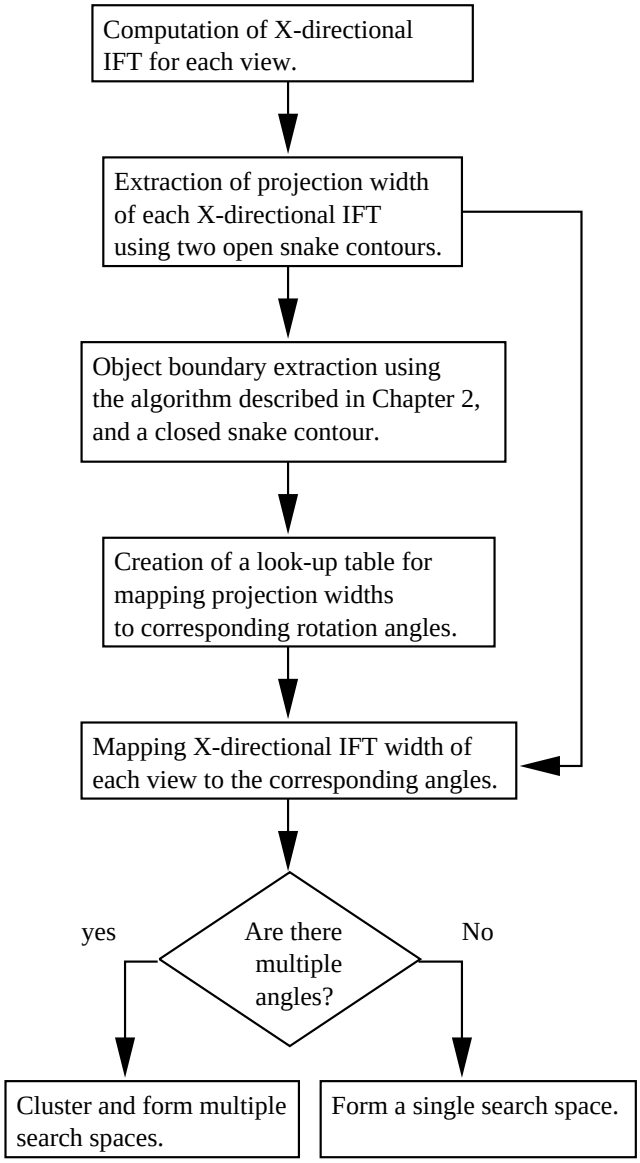


Figure 4.2: Block diagram of the algorithm to select the search space for each view.

4.3.1 Extraction of Projection Width from X-directional IFT

It was established in Chapter 2 that the width of the X-directional IFT equates to the Y-directional projection width w_l of the imaged object. In order to obtain an accurate estimate for w_l, the snake model [109] introduced in Chapter 2 is used with two open contours. The block diagram shown in Figure 4.3 indicates the steps involved in extracting the width w_l of X-directional IFT. As described

in Chapter 2 Section 2.4.2, the decision step shown as "Is the resulting contour satisfactory?" is performed by the user with the aid of the GUI.

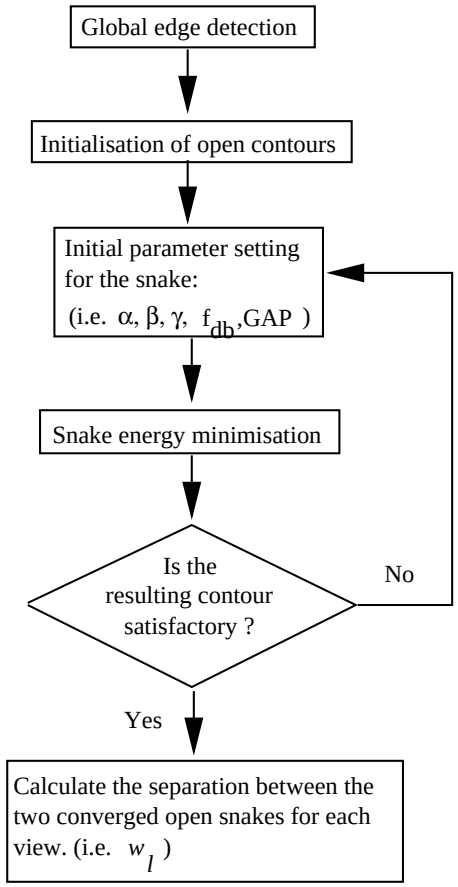


Figure 4.3: Block diagram of the algorithm for extracting the X-directional IFT projection width for each view.

In addition, a method of dynamic, linear interpolation is employed to facilitate snakes to sense the local details of the desired contour accurately or to capture the complicated shape (eg. sharp corners) of the desired contour properly. The parameter named GAP is the threshold for average length between two consecutive snake points, and can reflect the basic geometric property of the snake. Therefore, in order to avoid clustering or even looping, those snake points which are closer to the previous points within the value of GAP are removed. To avoid re-parametering after each interpolation and to maintain the continuity of optimizing iterations, the original parameter setting [109] at each contour point

is retained and each new contour point is given the parameter setting of its neighbouring points. In order to simplify the initialization of snakes, each initial contour is re-sampled prior to the energy minimization. Hence, the snake model can start from very simple initial conditions. Unlike the other interpolation schemes [128], this adaptable , dynamic interpolation scheme provides a simple control of basic geometric properties of snakes and encourage the best fit of the object contour without expensive computation.

In order to match the projection width w_l to an associated rotation angle, the outer boundary of the object have to be known *a priori* or should be extracted from the image with motion artifacts. This boundary is extracted using the algorithm described in Chapter 2. When the outer boundary of the object is established, a look-up table can be created, which matches the projection width (cast by this boundary on the x-axis) to the associated rotation angle. The method of creating this look-up table is described in the following subsection.

4.3.2 Look-up Table

A look-up table is created for Y-directional projection width versus rotation angle by rotating the estimated object boundary within the interval θ_{min} to θ_{max} in steps of $\Delta\theta$. θ_{min} and θ_{max} denote the probable minimum and maximum rotation angles for a known maximum angular span. The assigned values for θ_{min} and θ_{max} may vary according to the part of the anatomy which is being imaged and its freedom of in-plane rotation. In this chapter, $\theta_{min} = -90^o$ and $\theta_{max} = 90^o$ have been assigned to mimic head rotations. The value of $\Delta\theta$ determines the resolution of the estimated angle, however, small $\Delta\theta$ can lead to long estimation times. For the simulation experiments described in this chapter, $\Delta\theta$ is set to 1^o.

When this look-up table is created, the initial guesses for the search space of

rotation angles associated with each view can be established. A more detailed description of these initial guesses are given in the following section.

4.3.3 Initial Guesses

Since the look-up table contains rotation angles associated with each Y-directional projection width of $\hat{m}(x, y, k_m)$, it is possible to derive the best matching angles for each projection width from X-directional IFT. It should be noted that, in some cases where the scanned object displays geometric symmetries, there could be more than one possible initial guess. In this case, multiple search spaces are formed for each associated view. The angular span of the search space can be decided upon by the total angular span of the object and time constraints involved in the estimations. Typically, 10% of the total angular span can be adequate for accurate estimations. In the event of multiple initial guesses, some of the guesses may be contained within an already formed search space. In this case, the existing search space is extended rather than introducing a new search space, avoiding the possibility of overlapping search spaces.

The initial guesses are usually subjected to errors due to:

- errors in object boundary extraction,
- interpolation errors,
- errors in extraction of X-directional IFT projection width, and
- sub-pixel changes in projection width.

Therefore, the initial guesses only provide a rough estimate on where the correct rotation angle may be found. As an added precaution, an extra search space is formed at the estimated angle of the previous view. This search space is

dynamically allocated. In the event of a perfectly circular object, this method fails to identify any specific initial guesses. In such circumstances, the search space for test angles have to be expanded to the total angular span of the object, at the expense of estimation time.

4.4 Simulation Results

Simulation experiments are conducted using both a synthetic phantom and human head MR images shown in Chapter 1 Figure 1.11. The synthetic phantom is deliberately chosen to be symmetric in shape, in order to examine the limitations of the proposed algorithm and its viability in practice.

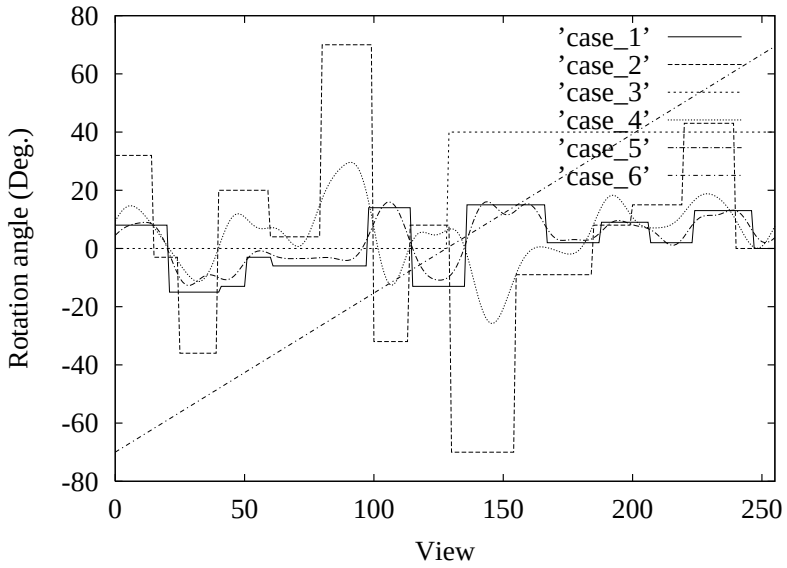


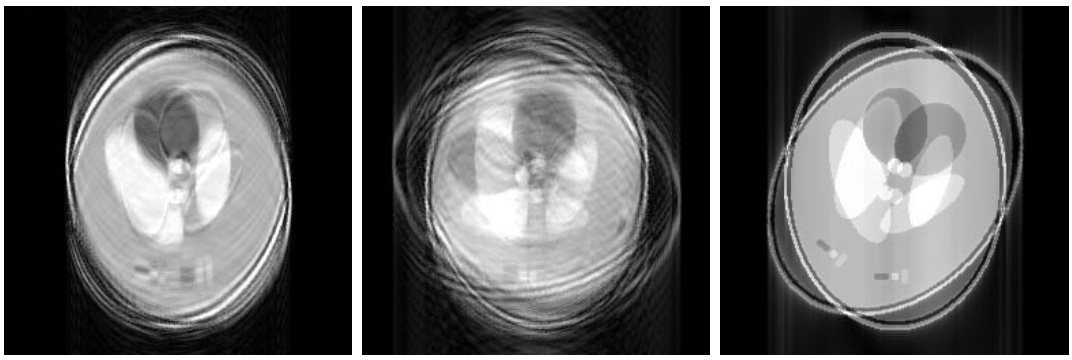
Figure 4.4: Rotational motion parameters for each phase encode for the six cases examined in the study.

Motion consisting of in-plane step-wise rotation as well as continuous rotation is used in the experiments. Simulated step-wise rotations are synthesized to contain worst case scenarios, such as sudden rotations near the $\left(\frac{N}{2}\right)^{th}$ view, large angle rotations and consecutive angle changes that do not incur any change in

the X-directional IFT width (w_l). The non-periodic continuous rotations are intended to represent worst case scenarios in practice. Rotation at constant angular velocity as analyzed in Chapter 3, is also used in the experiments to examine the performance of the algorithm when there are no significant overlap regions.

Figure 4.4 shows the view numbers and corresponding angles of rotation for six different cases studied. The first three cases and the sixth case involve the Shepp & Logan phantom [121], whereas cases 4 and 5 involve axial and sagittal MR images of a human head. The maximum angular span is $\pm 15^o$ for case 1, $\pm 70^o$ for case 2, 40^o for case 3, $\pm 30^o$ for case 4, $\pm 15^o$ for case 5, and $\pm 70^o$ for case 6.

(a) *(b)* *(c)*



(d) *(e)* *(f)*

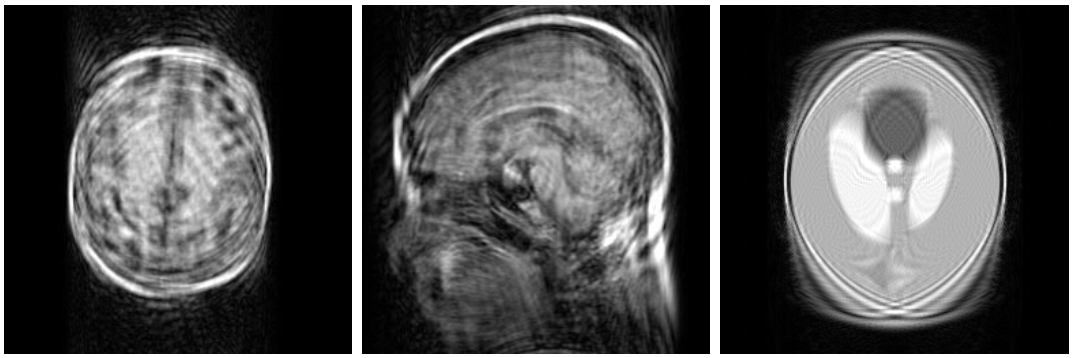


Figure 4.5: Reconstructed MR images from motion corrupted signals, using IFFT technique: *(a)* case 1 (MSE=1870.7); *(b)* case 2 (MSE=1989.5); *(c)* case 3 (MSE=1260.7); *(d)* case 4 (MSE=1147.2); *(e)* case 5 (MSE=2640.1); and *(f)* case 6 (MSE=2072.0) .

The reconstructed images for these six cases, using conventional 2D IFFT method [5], are shown in Figure 4.5. Severe motion artifacts are visible in all cases. Mean squared error value is calculated using Equation 4.8.

$$MSE = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [g(x,y) - m(x,y)]^2 \tag{4.8}$$

where $g(x,y)$ is the reconstructed image and $m(x,y)$ represents the original reference image. It should be noted that this measure will not be available in practice due to the unavailability of a reference image. However, this measure is included to quantify the performance of the proposed algorithm on the different experiments conducted in this study. The orientation of each object, when considered as the reference image $m(x,y)$ for each case is varied according to the ROI boundary extracted using the algorithm described in Chapter 2. These boundaries are shown in Figure 4.10, later in this Chapter. The most fitting orientation for a particular ROI boundary is used in determining the orientation of each object used for MSE calculations.

The X-directional inverse Fourier transforms for the six cases studied are shown as normalized images in Figure 4.6. Notice that for step-wise rotations (i.e. cases 1, 2 and 3), the projected width (w_l) changes abruptly whereas for continuous rotations (i.e. cases 4, 5 and 6) , the transitions are relatively smooth.

The initial open contours on the edge images of X-directional IFT images are shown in Figure 4.7. These contours are placed using the GUI of the contour extraction software, as described in Chapter 2 Section 2.4.2. The edge detectors used for this purpose are Sobel [129] (for cases 2, 4, 5 and 6) and Canny [130] (for cases 1 and 3). The choice of the edge detector is based on the user preference after visually examining the edge images using the GUI. However in this study, both the edge detectors are examined. The edge images shown in Figure 4.7 delivered a satisfactory snake contour within lower number of iterations. The

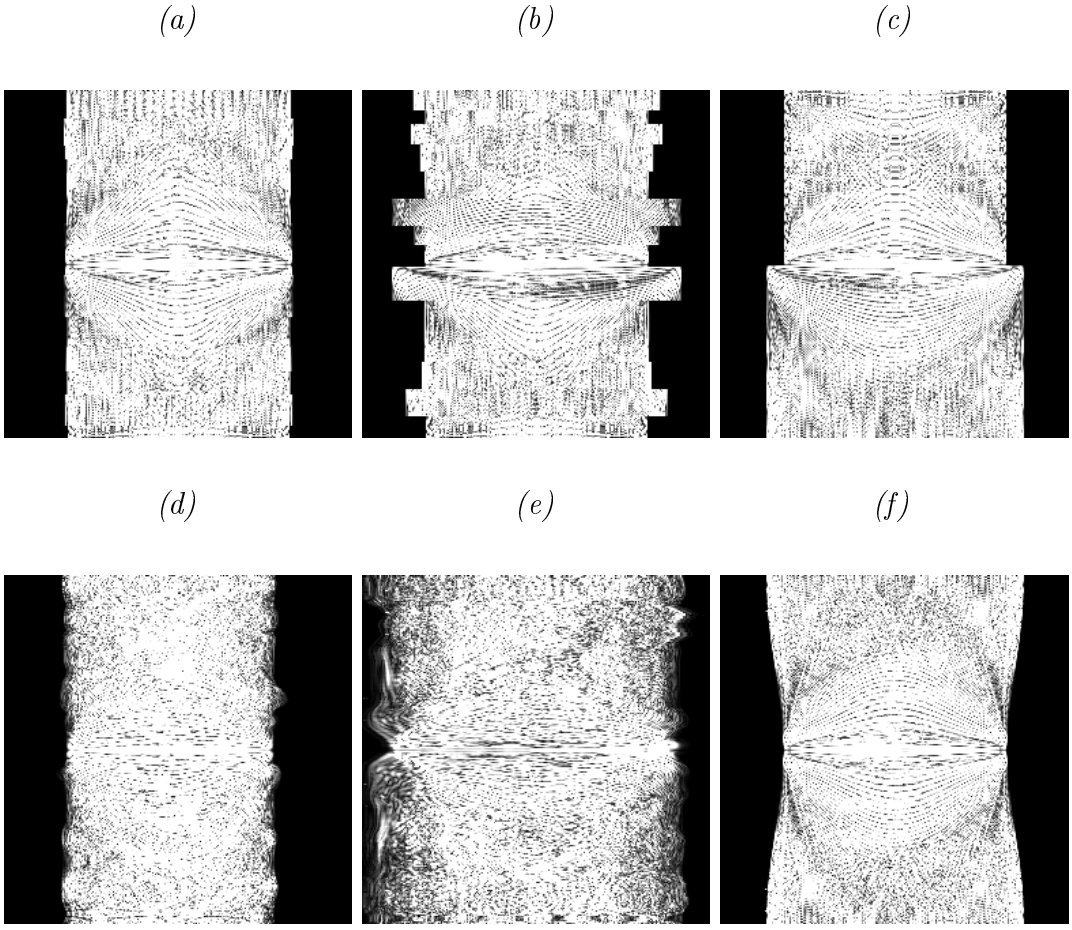


Figure 4.6: X-directional IFT images. Magnitude of the resulting values of each view is normalized to the interval [0,255]: *(a)* case 1 (step-wise); *(b)* case 2 (step-wise); *(c)* case 3 (step-wise); *(d)* case 4 (continuous); *(e)* case 5 (continuous); and *(f)* case 6 (constant angular velocity).

difference in the number of iterations was marginal. Hence, either of the two edge detectors can deliver a satisfactory solution.

The converged snake contours are shown in Figure 4.8. It is important to realize that the purpose of extracting the edges using snakes is to establish unbroken edges on both sides of the X-directional IFT images. As can be seen in Figure 4.7, this purpose cannot be fulfilled using simple edge detectors.

The snake parameters [109] used for each open contour and the iterations (It) involved are listed below for each case. These parameters are selected as described

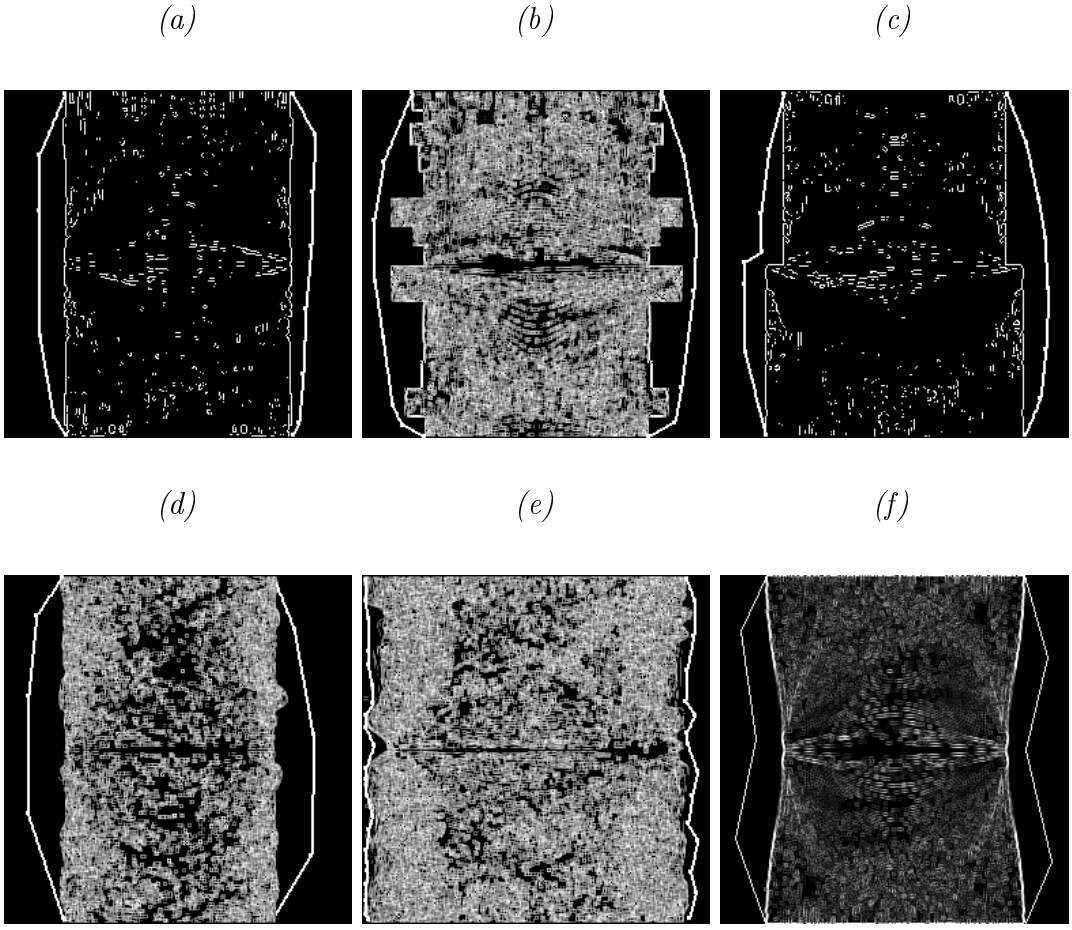


Figure 4.7: Initial snake contours placed on the edge images of the normalized X-directional IFT images shown in Figure 4.6. The edge detectors used are indicated for each case: *(a)* case 1 (Canny); *(b)* case 2 (Sobel); *(c)* case 3 (Canny); *(d)* case 4 (Sobel); *(e)* case 5 (Sobel); and *(f)* case 6 (Sobel).

in Chapter 2 Section 2.4.2. L represents the left edge whereas R represents the right edge:

- Case 1 - $L(\alpha = 0.6,\ \beta = 0.9,\ \gamma = 1.0,\ f_{db} = 0.8,\ GAP = 4$ and $It = 50)$, $R(\alpha = 0.6,\ \beta = 0.9,\ \gamma = 1.0,\ f_{db} = 0.8,\ GAP = 4$ and $It = 41)$.
- Case 2 - $L(\alpha = 0.5,\ \beta = 1.0,\ \gamma = 0.95,\ f_{db} = 1.15,\ GAP = 5$ and $It = 86)$, $R(\alpha = 0.5,\ \beta = 1.0,\ \gamma = 0.9,\ f_{db} = 1.0,\ GAP = 5$ and $It = 99)$.
- Case 3 - $L(\alpha = 0.6,\ \beta = 0.9,\ \gamma = 1.0,\ f_{db} = 0.8,\ GAP = 4$ and $It = 75)$, $R(\alpha = 0.6,\ \beta = 0.9,\ \gamma = 1.0,\ f_{db} = 0.8,\ GAP = 4$ and $It = 71)$.

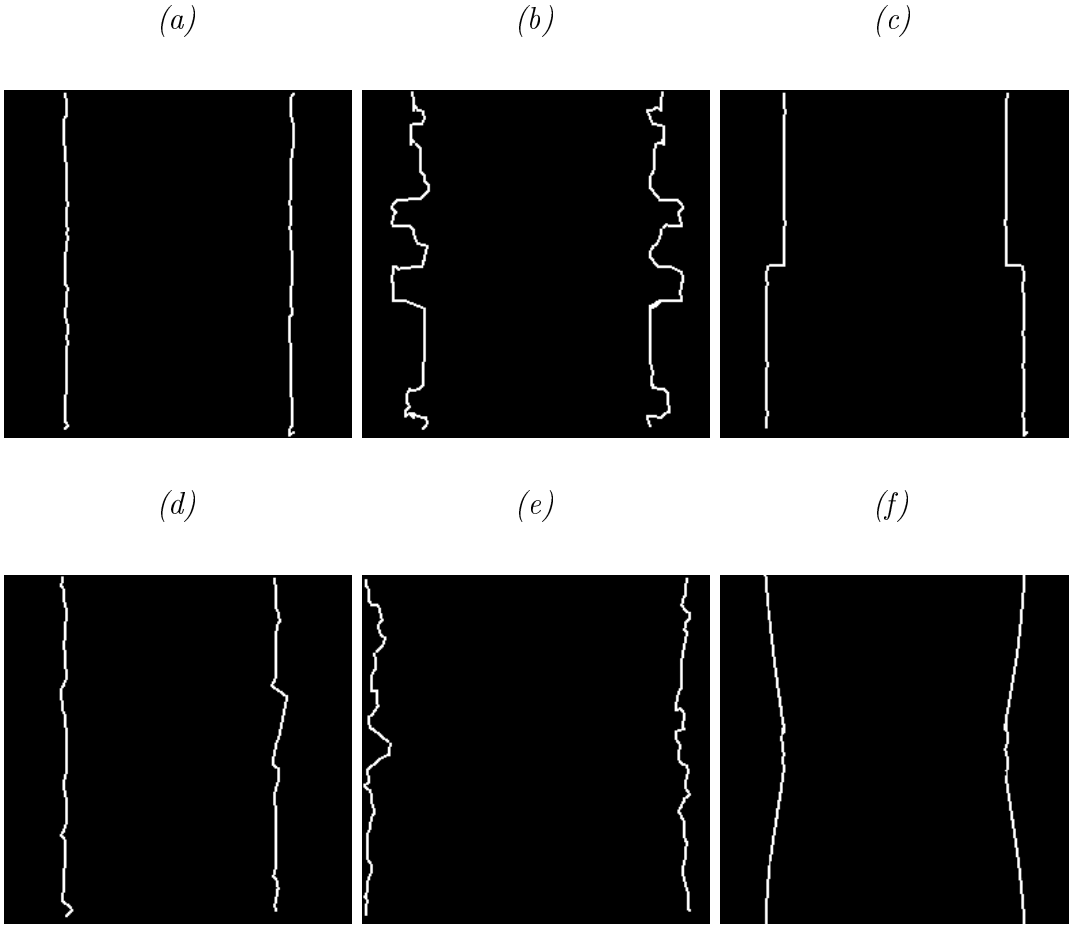


Figure 4.8: Converged snake contours for X-directional FT width extraction: *(a)* case 1; *(b)* case 2; *(c)* case 3; *(d)* case 4; *(e)* case 5; and *(f)* case 6.

- Case 4 - $L(\alpha = 0.9,\ \beta = 1.0,\ \gamma = 1.0,\ f_{db} = 0.6,\ GAP = 5$ and $It = 62)$, $R(\alpha = 0.9,\ \beta = 1.0,\ \gamma = 1.0,\ f_{db} = 0.75,\ GAP = 5$ and $It = 96)$.

- Case 5 - $L(\alpha = 0.7,\ \beta = 1.5,\ \gamma = 2.0,\ f_{db} = 1.3,\ GAP = 4$ and $It = 40)$, $R(\alpha = 0.7,\ \beta = 1.5,\ \gamma = 2.0,\ f_{db} = 1.3,\ GAP = 4$ and $It = 31)$.

- Case 6 - $L(\alpha = 0.6,\ \beta = 0.9,\ \gamma = 1.0,\ f_{db} = 0.8,\ GAP = 4$ and $It = 50)$, $R(\alpha = 0.6,\ \beta = 0.9,\ \gamma = 1.0,\ f_{db} = 0.8,\ GAP = 4$ and $It = 41)$.

The effect of using two straight lines at the left and right edges of the field of view for the initial snake contours was also examined. The result of such an initialization is shown in Figure 4.9 for Case 2. Since the open snake contour end points are fixed, the converged contour display errors towards these end points

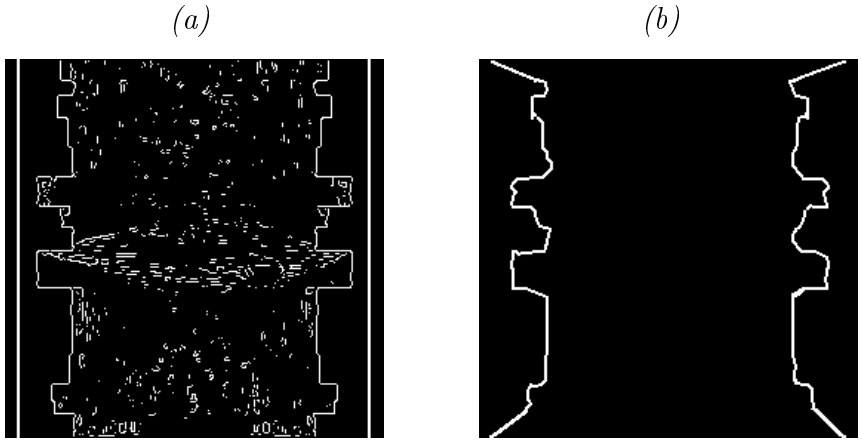


Figure 4.9: The effect of using straight line initial contours for extraction of X-directional FT width: *(a)* initial contours placed on the edge image (Canny) of Case 2; *(b)* converged snake contours. The snake parameters used are equivalent to the ones listed above for Case 2 for both the left and right edges.

(see Figure 4.9(b)). However, these errors are associated with views containing high spatial frequencies. It is later shown in this section, that the rotation parameters associated with these views are usually discarded due to low reliability. Therefore, such an initialization will not significantly affect the final result.

As described in Chapter 2, the object boundary extraction is performed using closed snake contours. The converged closed snake contours are shown in Figure 4.10. These contours are used to construct the look-up table described in Section 4.3.2. Using this look-up table, the initial search spaces are derived for the motion estimation algorithm. Objects with approximate geometric symmetries produce multiple search spaces for some views. This results in unnecessary elongation of the estimation times. In order to expedite the estimation process, only the search space, which contained the previous estimated angle is considered. This effectively reduces the estimation time by half with good results for continuous rotations. However, the algorithm fails to track sudden angle changes involved in step-wise rotations. As a solution to this problem, the minimum slew

rate among the initial guesses are computed. At large variations in rotation angles, the high slew rate provides a guide for searching all possible initial guesses, providing better tracking of sudden angular changes.

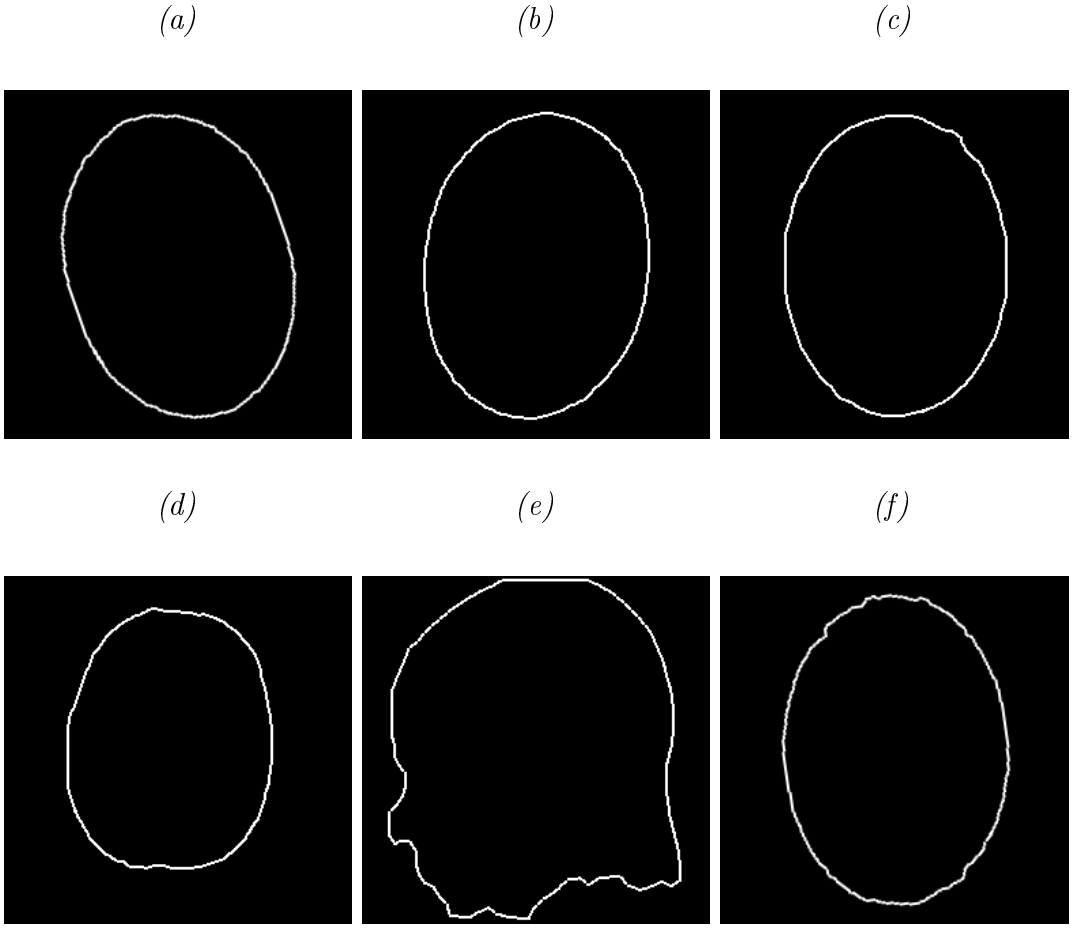


Figure 4.10: Converged snake contours for object boundary extraction. The snake parameters [109] used for each closed contour and the iterations (It) involved are indicated for each case: *(a)* case 1 ($\alpha = 0.7$, $\beta = 1.6$, $\gamma = 2.0$, $f_{db} = 0.9$, $GAP = 4$ and $It = 147$); *(b)* case 2 ($\alpha = 0.7$, $\beta = 1.3$, $\gamma = 1.8$, $f_{db} = 0.8$, $GAP = 4$ and $It = 55$); *(c)* case 3 ($\alpha = 0.7$, $\beta = 1.4$, $\gamma = 1.8$, $f_{db} = 0.8$, $GAP = 4$ and $It = 60$); *(d)* case 4 ($\alpha = 0.7$, $\beta = 1.6$, $\gamma = 2.0$, $f_{db} = 0.8$, $GAP = 4$ and $It = 89$); *(e)* case 5 ($\alpha = 0.8$, $\beta = 1.2$, $\gamma = 1.5$, $f_{db} = 1.2$, $GAP = 7$ and $It = 56$); and *(f)* case 6 ($\alpha = 0.7$, $\beta = 1.6$, $\gamma = 2.0$, $f_{db} = 0.9$, $GAP = 4$ and $It = 132$).

Since the search space near the initial guesses is fixed at an arbitrary value (10% of the total angular span) for all angle estimations, it is found that in some cases, the correct rotation angle falls outside the search space. Such events are

accompanied by estimations with unexpectedly low membership values. Therefore, if the membership value of a particular estimation falls below 40% of the previous estimation, a re-estimation is invoked for that particular view with a doubled search space near the initial guess. This is necessary to track the motion accurately, although it requires up to 4 times the usual estimation time. At high spatial frequencies, the signal to noise ratio (SNR) of data is found to be inadequate to give acceptable membership values for the angle estimations. Therefore, those estimates with membership value less than a pre-defined α-cut [115] are discarded. The value of the α-cut was defined as 80% of the average membership of all the estimated parameters. Discarding these unreliable estimates causes little effect on the final reconstructed image since noise corrupted data hardly give additional information to upgrade the quality of the image. Where possible, the rotation angles of discarded estimations are derived using linear interpolation of nearby valid estimations. The results of angle estimations (i.e. after imposing the α-cut) for the five experiments considered, are shown in Figure 4.10. The corresponding membership values are plotted in Figure 4.11. The shift in the estimated angles from the actual motion is due to the fact that the algorithm considers the rotation angle of the object at the acquisition of $\left(\frac{N}{2}\right)^{th}$ view to be 0^o. This does not affect the artifact suppression scheme, however, the corrected image displays a rotated version of the initial object. Root mean squared error (RMSE) of the estimations was computed disregarding the angular shift.

In Case 2, high incidence of data overlap regions due to large angle rotations result in more reliable angle estimations, especially at the high frequency views, compared to small angle rotations with less overlap (eg. Case 1 and Case 6).

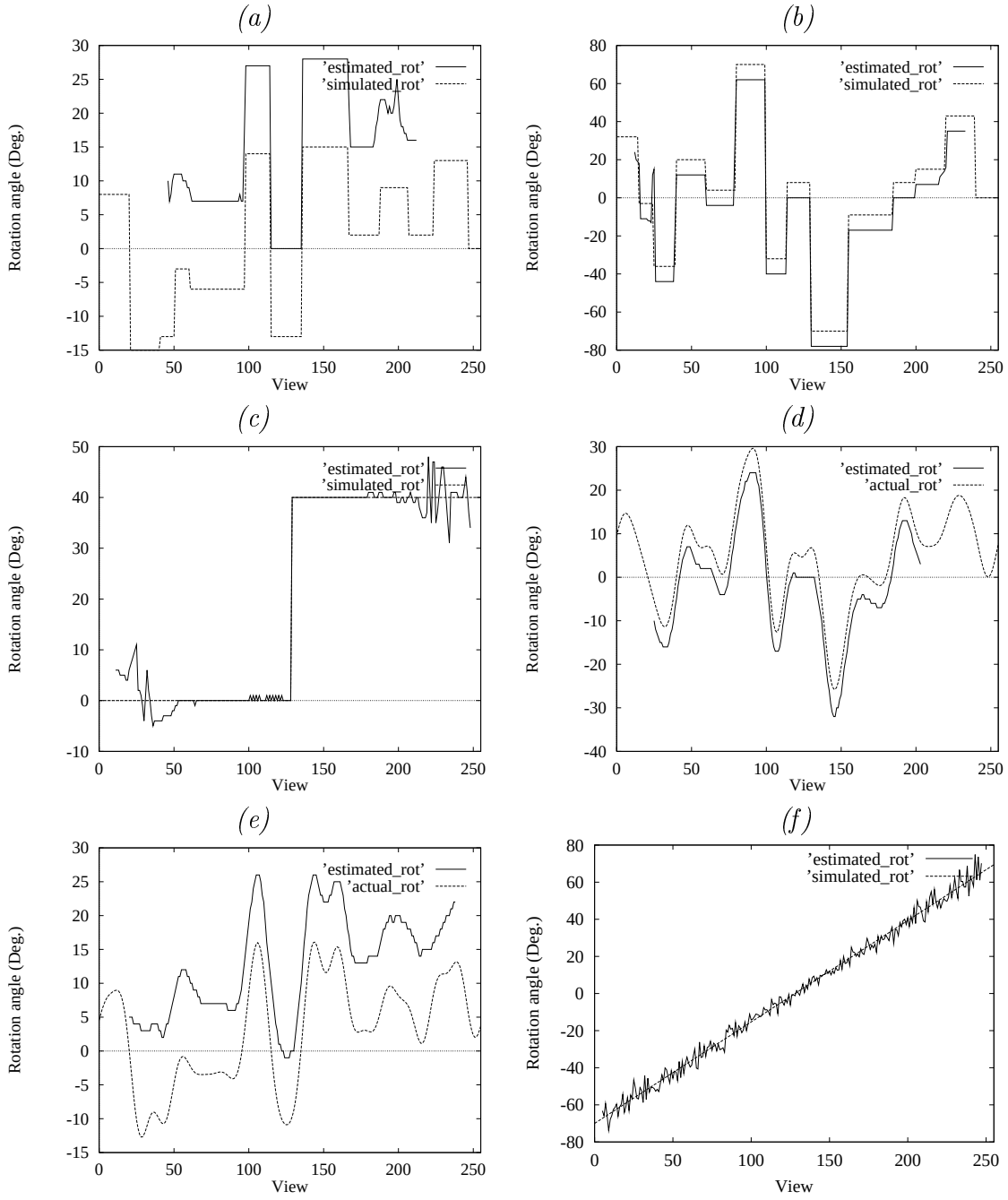


Figure 4.11: Angle estimation results - Estimated rotation angles (solid line) are compared to the simulated rotation (dashed line): *(a)* case 1 (RMSE=2.388^o for full span of 30^o); *(b)* case 2 (RMSE=5.718^o for full span of 140^o); *(c)* case 3 (RMSE=2.544^o for full span of 40^o); *(d)* case 4 (RMSE=0.512^o for full span of 55^o); *(e)* case 5 (RMSE=1.911^o for full span of 27^o); and *(f)* case 6 (RMSE=10.061^o for full span of 140^o).

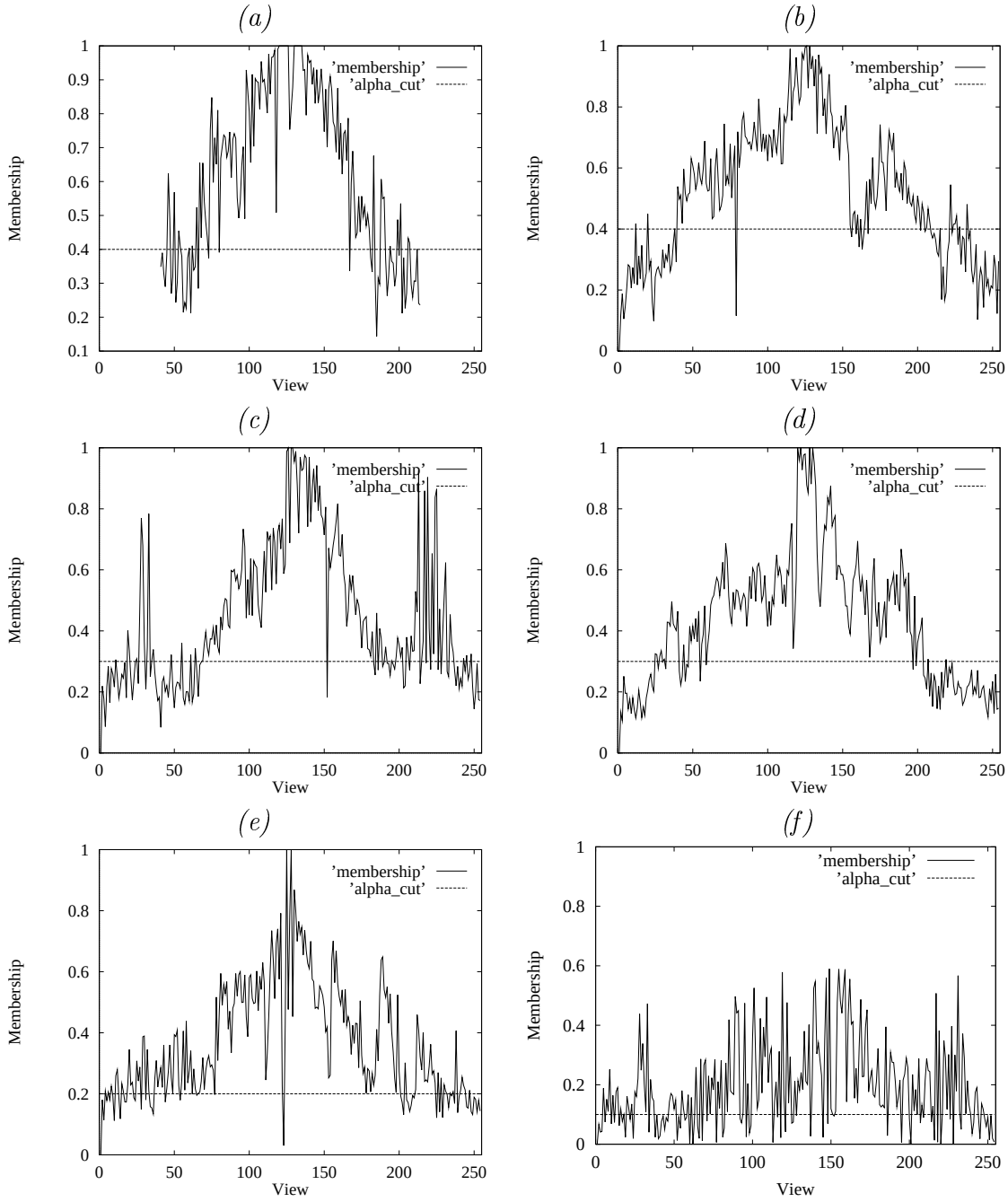


Figure 4.12: Angle estimation results - Membership values (solid line) representing the reliability of each angle estimate are shown with the α-cut value (dashed line) used for choosing the valid estimates: *(a)* case 1; *(b)* case 2; *(c)* case 3; *(d)* case 4; *(e)* case 5; and *(f)* case 6.

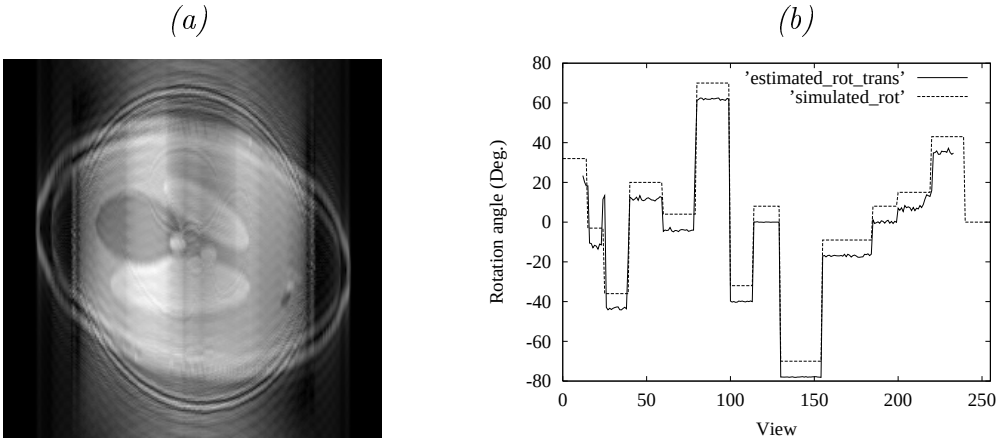


Figure 4.13: Rotation angle estimation results with concurrent translational motion, for Case 2: *(a)* Reconstructed image using IFFT (MSE = 2871.3); *(b)* Estimated rotation angles for motion involving rotation with concurrent translations (solid line) compared to the simulated rotation (dashed line) ($RMSE$ = 9.071 for full span of 140^o).

In order to examine the performance of the proposed algorithm under concurrent translational motion, translational motion effects are added to Case 2. The X-directional and Y-directional translations are computed using Equations 4.9 and 4.10 respectively, for the ${k_y}^{th}$ phase encode.

$$p(k_y) = 5\sin\left(\frac{\pi k_y}{100}\right) \tag{4.9}$$

$$q(k_y) = 2\sin\left(\frac{\pi k_y}{100}\right) \tag{4.10}$$

The reconstructed image using conventional IFFT is shown in Figure 4.13(a), whereas the rotation angle estimation results are shown in Figure 4.13(b). The RMSE value for this estimation was 9.071 with respect to the shift corrected simulated rotations, approximately 3.3 higher than the similar estimation for data without concurrent translational motion. Therefore, it is observed that concurrent translational motion can degrade the accuracy of the rotational angle estimations. However, the algorithm is still capable of producing reasonable results as shown in Figure 4.13(b).

The total estimation times for continuous rotations range from 132 minutes to 157 minutes on a Sun SPARC 2 workstation, whereas step-wise rotations incur a maximum of 167 minutes. The total number of valid estimations is in excess of 190 angles. Compared to previously published work [78] [81], which required 4.5 hrs for 10 estimations [78], the proposed algorithm performs efficiently and avoids an exhaustive search on all possible combinations of angles [81].

The estimated angles are then used to suppress the rotational motion artifacts using the algorithms described in the following chapters.

4.5 Summary

Extraction of in-plane rotation information directly from the acquired MR data is possible. The results indicate that the proposed method is capable of estimating rotation angles corresponding to each acquired view without limitations on the type of in-plane rotational motion involved, at a fraction of the time reported in previous studies. Such a reduction in estimation time was mainly due to the transformation of the problem from N-dimensional optimization in to N$\times$(1-dimensional) optimizations. An interesting observation is that the proposed method performs particularly well for large angle rotations, where there is an abundance of data overlap. Therefore, this method of motion estimation can be very useful in cases where previously proposed methods are either ineffective or computationally prohibitive.