THE UNIVERSITY OF CALGARY

An Image Processing Approach for Motion Artifact Suppression in Magnetic Resonance Imaging

by

Weifang Yang

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA

November, 1998

0-612-35028-2

ABSTRACT

In this thesis. post-processing methods for motion artifact suppression in magnetic resonance imaging (MRI) are investigated. Based on experimental observations. a generalized motion model is empirically proposed. The non-rigidity or spatially variant characteristics of motions are taken into account by introducing a distortion transfer function (DTF). A technique to estimate DTFs directly from corrupted images is developed. With DTFs identified. a composite image processing method is proposed to correct ghost artifacts caused by non-rigid periodic motions along the slice selection axis. There are several image processing tasks involved in this composite method. including contour detection and contour-based region labeling. It is proposed that contour detection is done by a new technique: the snake. The generalized motion model and the composite image processing method are demonstrated for both phantom images and an abdomen image with unknown respiratory motion.

ACKNOWLEDGEMENTS

First I would like to thank my supervisor Dr. M. R. Smith for providing me the opportunity to complete this thesis. His countless hours of help and direction, his patience. his encouragement and excellent technical writing tips are very much appreciated. I gratefully acknowledge the Natural Sciences and Engineering Research Council (NSERC) of Canada, University of Calgary and Department of Electrical and Computer Engineering at University of Calgary for their financial support. Many thanks to L. Chen. S. Krishnan. F. Faghih, T. Mathews and R. Gokaraju for giving me help about using some programs I needed during this thesis project and thesis writing. Many thanks to J. Zeng, with whom I had some helpful conversations about this thesis.

I would like to thank my husband Ning for his love, support and encouragement. My sincere gratitude to my parents, my brother and all those who are supportive in my life. Thanks to the Kowalski family for their hospitality. kindness and love.

To the memory of my grandfather

CONTENTS

CHAPTERS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Thesis Subject

Magnetic resonance imaging (MRI) has displayed great potential in medical imaging since it was introduced in the 1970's [Wehrli et al. 1988; Slichter 1990]. Compared with other biomedical imaging techniques, such as ultrasound imaging and x-ray computerized tomography, MRI has the following superior properties [Wehrli et al. 1988; Kak and Slaney 1988; Shung et al. 1992; Hamilton 1982]: MRI can provide high contrast images; MRI can easily select an imaging plane in any direction of a human body; MRI is considered safer than x-ray related imaging techniques.

However conventional MRI techniques, especially two-dimensional Fourier imaging technique, need a long data acquisition time to collect all the data required to reconstruct a high quality image. During the long data acquisition time, physiological movements and random body movements of the imaged subject may occur. The subject's movements during this scan time affect the acquired data and result in motion artifacts which degrade the reconstructed image.

How to effectively suppress artifacts caused by motions has been an active topic in MRI. The existing techniques can be categorized as :

- techniques that directly restrain the motion itself.
- techniques that modify the conventional data acquisition procedures.
- fast imaging techniques.
- image processing (or post-processing) techniques.

Among these methods, post-processing techniques are very attractive as they require no changes in the conventional MRI system, nor in the imaging method, nor in the data acquisition procedures. The project for this thesis has been focused on the post-processing techniques for motion artifact suppression in MRI. The subjects of the project are two:

1. to investigate a mechanism and the corresponding mathematical model of how artifacts are introduced
2. to develop post-processing algorithms to suppress artifacts caused by motions.

In this chapter. the basics of MRI are very briefly presented. The effects of motions on MRI measured k-space data and reconstructed images are then analyzed. The existing methods of motion artifact suppression in MRI are categorized. Finally the outline of this thesis is described.

1.2 Basics of Magnetic Resonance Imaging (MRI)

1.2.1 Spin Magnetization

The physical background of MRI is nuclear magnetic resonance (NMR). which is an interaction between the applied magnetic field and nuclear magnetization of atoms of the imaged object [Wehrli et al. 1988; Slichter 1990].

The nucleus of an atom has a positive charge and is considered as spinning about an axis. This spinning nucleus generates a magnetic dipole. which is similar to that of a bar magnet with its direction parallel to the axis of the spinning. Normally when there exists no external magnetic field around the object. the dipoles of nuclei take random directions and cancel out each other. As a whole. the object then shows no magnetization. However when a static external magnetic field B_0 (called a DC magnetic field in some books) is applied to the object. the nuclei interact with the field via their dipoles. The dipoles rotate about the direction of the static field B_0 at a very specific resonance frequency ω. The frequency ω is directly proportional to the magnitude of the static field B_0. That is

$$\omega = \gamma B_0$$

where γ is a constant called the magnetogyric ratio. This resonance frequency ω is called the Larmor frequency.

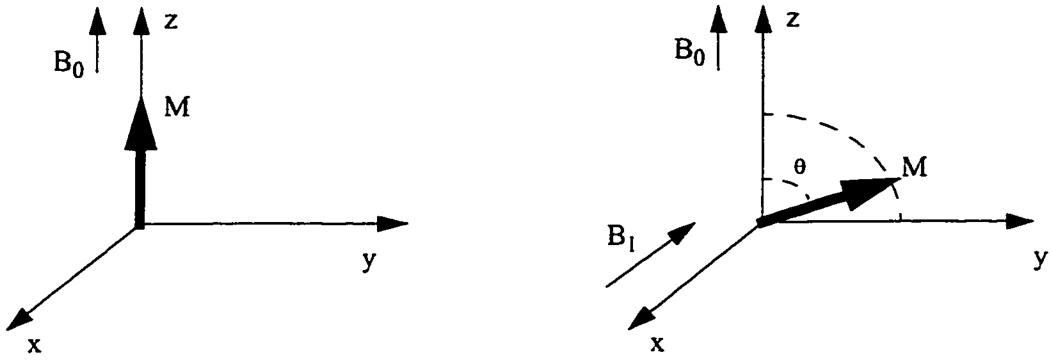
A vector model is widely used in the literature to visualize NMR. In this model. nuclear magnetization is represented as magnetization vectors [Wehrli et al. 1988: Slichter 1990]. We assume the magnetic field B_0 is in the direction of the z axis. Then the magnetization $\vec{M}$ of the atoms is a vector along the z axis as shown in Fig. 1.1(a). We also assume that the x and y axes rotate about the magnetic field axis (z axis) at the Larmor frequency. Therefore when a second external magnetic field B_1 of the Larmor frequency. and in the direction of the x axis. is applied. the magnetizaton $\vec{M}$ is perturbed away from the z axis toward the xy plane by an angle θ as shown in Fig. 1.1(b). The angle θ depends on the value and duration τ of the second external magnetic field B_1 as :

$$\theta = \gamma B_1 \tau$$

The magnetization M_z along z axis is reduced and simultaneously a xy plane magnetization (also called transverse magnetization) M_{xy} is produced.

The Radio Frequency (RF) pulse required to rotate the nuclei by 90^o is called a 90^o RF pulse. The magnetization is along the y axis after a 90^o RF pulse is applied. Similarly. an RF pulse capable of rotating the nuclei by 180^o is called an 180^o RF pulse and places the corresponding magnetization along the $-z$ axis.

According to the quantum theory, this process of magnetization perturbation is a process of energy resonance absorption and is an excitation process. In the time following excitation, the magnetization wants to return to its equilibrium after giving off energy by a number of relaxation mechanisms.



(a) The net magnetization of atoms is along the z axis when the object is placed in a static magnetic field B_0 in the z direction

(b) Application of B_1, the second external magnetic field at the Larmor frequency tips the magnetization away from the z axis toward the xy plane by an angle θ

Figure 1.1. Vector representation of the spin magnetization in a rotating frame of reference

1.2.2 Relaxation

There are two distinct nuclear relaxations that occur following excitation. The first is related to an exponential recovery of the magnetization M_z along the z axis to its equilibrium values. The second relaxation is associated with an exponential decay in the transverse magnetization M_{xy}.

The first relaxation is called spin-lattice relaxation as it is the loss of energy from the pulse to the surroundings (lattice) as thermal energy. The exponential time constant is called T_1, and describes the rate of return of the M_z magnetization to its equilibrium value.

The second relaxation is called spin-spin relaxation. The corresponding exponential time constant is T_2. Normally $T_2 \leq T_1$ as the transverse magnetization M_{xy} decays more rapidly than M_z recovers. The signal detected by a nuclear magnetic resonance (NMR) system is proportional to the transverse magnetization M_{xy}. Hence the detected signal following an excitation RF pulse dies away as a result of the T_2 relaxation.

1.2.3 Principles of Magnetic Resonance Imaging (MRI)

Nuclear magnetic resonance as a technique has been used by scientists since the mid 1940s. However magnetic resonance imaging (MRI) for medical use did not become available until 1970s. The key to MRI is encoding the spatial information into the detected NMR signal. One of the most popular MRI techniques is Two-dimensional (2D) Fourier MRI, which is very elegant in theory [Wehrli et al. 1988].

The foundation of the 2D Fourier MRI technique is the use of field gradients. Usually three field gradients G_x, G_y and G_z, in the x, y and z directions respectively, are employed. A field gradient G_z along the z axis is used to select a cross section to be imaged. The z axis is also called the slice selection axis. The field gradient G_y

along the y axis is used to encode the spatial information along the y axis into the phase of the NMR signal. At the end of the period (time t_y) for which G_y is applied. the relative phase of a signal from a nucleus depends on the gradient strength G_y and the nucleus position y along the y axis as:

$$\phi = \gamma y \int_0^{t_y} G_y(t)dt = \gamma G_y y t_y \propto y$$

The field gradient G_x along the x axis is used to encode the spatial information along the x axis into the frequency of the NMR signal. When a linear gradient G_x is superimposed on the main magnetic field B_0. the Larmor frequency ω depends on position along the x axis as:

$$\omega = \gamma(B_0 + xG_x)$$

A series of n data points of the NMR signal is recorded in the presence of this G_x. We repeat the procedure with a different value of the gradient G_y. and record a new set of n data points. which differs from the first set in the initial phases. Repeating the procedure for m different values of G_y produces a data set consisting of m rows and n columns. The variable along the row is the time and the variable down the columns is the phase step. Hence the x axis is called the frequency-encoding axis or read-out axis. The y axis is termed the phase-encoding axis.

In summary, the basic procedure for applying the magnetic fields for generating an m by n data set during 2D Fourier MRI is:

1. Select a cross section to be the imaged plane by using the z direction field gradient G_z.

2. Apply a phase-encoding gradient G_y for a fixed time.

3. Apply a read-out gradient G_x and collect n data points.

4. Change the value of phase-encoding field gradient G_y and repeat steps 1, 2 and 3 a total of m times.

The effect of the applied RF pulses will be discussed in the next paragraph.

Note that the time required for step 1, 2, and 3 is a few tens of millisecond. However the physics of the spin relaxation demands that excitation using the 90^o pulse can only be repeated on the time scale of a second in order to avoid complete saturation of the nuclei and as associated loss of signal intensity. The time between successive excitations, called T_R, must therefore be of that order. Since there are m phase-encoding steps, the total time required for an imaging procedure is $m \cdot T_R$. That is where this MRI data acquisition takes time.

Fig. 1.2 shows a commonly used spin-warp imaging sequence of 2D Fourier MRI [Wehrli et al. 1988]. The spin-warp sequence works as follows:

1. The 90^o selective RF pulse and field gradient G_z are applied to select a cross-section of the subject as the imaging plane. As a result, all the nuclei in this plane are excited. The nuclei out of this plane do not resonate to this RF pulse and will not contribute to the NMR signal.

2. The time-reversal gradient $-G_z$ is used to remove the unwanted effect introduced by the nonuniformity of the magnetic field. At the end of the time-reversal gradient, a much larger signal is achieved than that obtained without the reversal gradient.

3. The frequency-encoding field gradient G_x and the phase-encoding field gradient G_y are applied to encode the spatial information in the imaging plane into frequency and phase respectively.

4. The 180^o RF pulse is used to produce a spin echo and the G_x read-out pulse is used for reading out n data points of the NMR signal.

5. Change the phase-encoding field gradient G_y and repeat step 1, 2, 3, and 4 a total of m times.

6. An m by n data set is obtained.

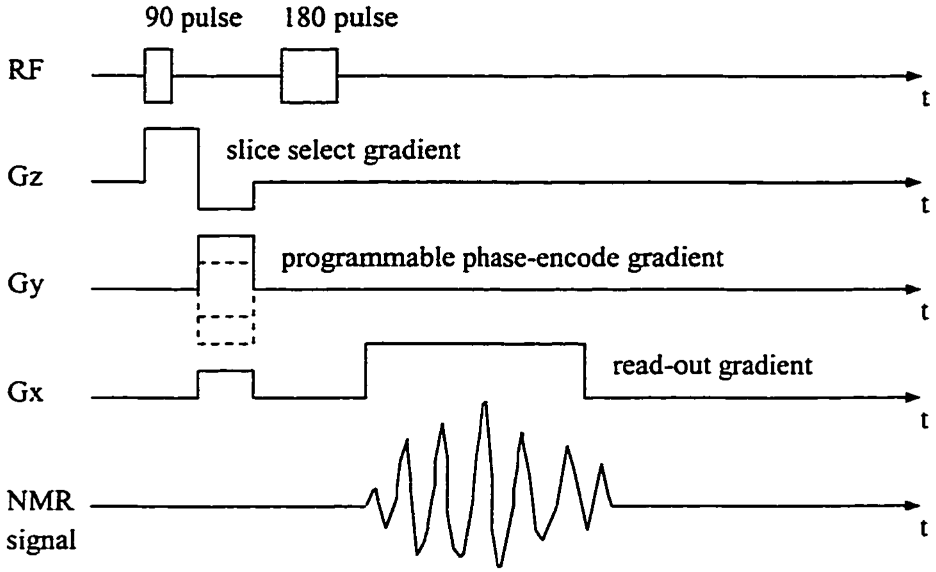


Figure 1.2. The field gradients and RF pulses associated with the spin-warp imaging sequence of 2D Fourier MRI [Wehrli et al. 1988]

1.2.4 K-space Data

Let's discuss the recorded m by n data set in step 6 in the above section.

Assume the spin density of the imaged cross section at location (x, y) in the imaging plane is $\rho(x, y)$ and the phase-encoding field gradient G_y at the m^{th} phase-encoding step is chosen to be proportional to the phase-encoding step m as: $G_y(m) \propto m$. If the imaged subject is stationary and if the relaxation effects are neglected, the NMR signal $s(t, m, x, y)$ from location (x, y) during this phase-encoding step m is given by:

$$s(t, m, x, y) \propto \rho(x, y) \exp(-j2\pi xt) \exp(-j2\pi my)$$

where $\exp(-j2\pi xt)$ represents the frequency term and $\exp(-j2\pi ym)$ is the phase term [Wehrli et al. 1988].

If the selected imaging slice is thin enough, the detected NMR signal $s(t,m)$ during phase-encoding step m is the summation of all the NMR signals from all locations in the imaging plane as:

$$s(t,m) \propto \int_y \int_x s(t,m,x,y)dxdy \propto \int_y \int_x \rho(x,y) \exp(-j2\pi xt) \exp(-j2\pi my)dxdy \tag{1.1}$$

The recorded n data points are given by:

$$s(n,m) = s(t,m)|_{t=n\Delta t} \propto \int_y \int_x \rho(x,y) \exp(-j2\pi nx) \exp(-j2\pi my)dxdy \tag{1.2}$$

Mathematically the right hand side of Equation 1.2 is the sampled version of the 2D Fourier transform of $\rho(x,y)$, the spin density image. Thus the recorded data $s(n,m)$ at phase-encoding step m is mathematically a row of the 2D Fourier transform of the imaged cross section. On the other hand, the recorded data itself is a time series physically. Therefore the recorded data $s(n,m)$ has both time and frequency properties. It is called k-space data. The elegance of 2D Fourier MRI is that an m by n data set of the image $\rho(x,y)$ can be reconstructed by doing 2D inverse discrete Fourier transform according to Equation 1.2 [Wehrli et al. 1988].

Note that this measured k-space data has complex values. Furthermore it is contaminated with noise. Thus the reconstructed image data set is also complex valued rather than real valued. So when we talk about actual MRI k-space data and reconstructed MRI images, we are talking about complex data sets. We can manipulate the k- and image space data sets as complex values prior to the final magnitude image being displayed.

1.3 Effects of an Object's Motion on Its K-Space Data and Reconstructed Image

A question to ask is how motion of a MR imaged object affects the reconstructed images and what do the resulting artifacts look like? This problem has been investigated from different points of view: experiment. physical theory and mathematical analysis by Schultz *et al.* [1984]. Wood and Henkelman [1985] and Xiang and Henkelman [1993].

Schultz *et al.* [1984] experimentally investigated the effects of motion on two-dimensional Fourier MRI. The phantom material with simulated respiratory motions and volunteers' abdominal walls with respiratory motions were investigated. The reconstructed images showed that the respiratory motions introduced both ghosts and blurring. By ghost we refer to the artifact that has similar structure of the true image. Blurring represents the smearing that makes the image indistinct.

Wood and Henkelman [1985] studied the problem theoretically and experimentally. The reconstructed image of a single point was discussed based on the physical background. The analytical formula of this point spread function (PSF) showed that. even without motion, the PSF is no longer a Delta function. due to the apodization and the discrete and finite sampling. Wood and Henkelman [1985] also studied an example when there existed a periodic translational motion in the x direction. The PSF was shown to have repetition terms with a uniform separation y_G in phase-encoding direction y. Image ghosts resulted from these repetition terms. Each repetition term was amplitude modulated differently. The amplitude modulation of each term resulted in blurring. The distance between two consecutive ghosts, y_G, was determined by:

$$y_G = \frac{2\pi F T_R}{\gamma \Delta G t_y} \tag{1.3}$$

where F is the frequency of the periodic motion. T_R is the time interval between two consecutive phase-encoding steps. γ is the nuclear gyromagnetic ratio. ΔG and t_y

are the monotonical increment and duration of phase-encoding field gradient G_y.

According to Equation 1.3, the following conclusions can be drawn :

- The higher the motion frequency F, the wider the spacing between two consecutive ghosts;
- Decreasing either t_y, the duration, or ΔG, the step size of the phase-encoding gradient increases the ghost separation.

Wood and Henkelman [1985] also did experiments with the phantom of an isolated point. The phantom point was controlled to move with different motion frequencies and motion amplitudes in the x direction during the data acquisition. The PSF images were reconstructed from the received MRI data. In experiments, the ghosts and blurring displayed the same properties as theoretically predicted when the motion parameters changed.

Xiang and Henkelman [1993] presented the following conceptual framework to view the effects of motions on k-space data when only inter-view motions are considered. Assume the imaged slice is selected to be the xy plane at z_0 by manipulating the z direction field gradient G_z. The slice at the fixed spatial location z_0 changes with time due to the motion of a moving (dynamic) object, compared to a stationary one. This means the slice at z_0 at the time when the i^{th} row is recorded is different from that at the time when the next row, $(i+1)^{th}$ row, is recorded. Thus the selectively excited slice can be considered as a 2D object whose intensity and shape are changing with time. These changes can be described by a stack of "movie frames" separated by the phase-encoding repetition time T_R. This conceptual framework is very useful to get the compounded k-space data when the relationship between the movie frames is not regular but arbitrary. According to this conceptual framework, the recorded rows at different phase-encoding steps are associated to different 2D objects (or 2D frames).

After the effects of motions are studied, the next question is how to remove the artifacts caused by motions. This is topic of the next section.

1.4 The Categories of Motion Artifact Suppression Methods in MRI

Generally there are four main types of motion involved in MRI [Wehrli et al. 1988]:

- respiratory motion.
- cardiac motion.
- fluid motion. such as blood flow.
- random patient motion.

Usually the artifacts caused by these motions is complex combinations of ghosts and blurring. Various methods to suppress artifacts caused by different motions have been proposed since early 1980s. From a literature search we have categorized existing methods as:

- methods that directly restrain the motion itself.
- methods that modify data acquisition procedures.
- fast imaging methods.
- post-processing methods.

1.4.1 Methods that directly restrain the motion itself

A direct, simple and effective way to suppress motion artifacts in MRI is, if possible, to suppress the motion itself during the data acquisition time [Wehrli et al. 1988]. For example, in order to remove artifacts caused by respiratory motion, the breath holding method can be used. which requires the imaged subjects to hold their breath during the scan time.

However patients do not feel comfortable with the method. Sometimes the method fails due to lack of cooperation from the imaged subjects, such as infants and severely ill patients. Moreover. the involuntary motions, such as blood flow. are impossible to hold.

1.4.2 Methods that modify the data acquisition procedures

By modifying the data acquisition procedure, you can choose the way of acquiring data to suppress artifacts caused by certain motions. MRI radiologists and technicians have experimented with a variety of techniques. The gating method. the averaging method. the deliberate ghost positioning method and the ROPE method [Schultz et al. 1984: Wood and Henkelman 1986; Macgowan and Wood 1996: Constable 1997; Langenberger and Moser 1997; Kruger et al. 1997] are discussed below in detail.

1.4.2.1 The gating or synchronizing method

Gating or synchronizing is frequently used in MRI [Schultz et al. 1984: Wood and Henkelman 1986]. The idea is that for periodic motions. an attempt is made so that every view of k-space data is recorded at the same phase of the motion period.

Respiratory gating is done by monitoring the respiratory cycle. As the chest dimensions are observed to be more constant during the end expiration, only those signals acquired during the end expiration are used to reconstruct the image.

Similarly, cardiac gating is accomplished by initiating the MR imaging sequences according to the R wave of the electrocardiagram. The cardiac volume is relatively constant during approximately 50 msec immediately following the R wave. The phase-encoding repetition time is determined by the heart rating.

The method of combining the cardiac gating with respiratory gating is accomplished by initiating the MR imaging sequence 1 msec after the R wave that occurs during the end expiration. The quality of the reconstructed image is improved very much by this composite method. However, the problem of the gating methods is that the data acquisition time increases excessively.

1.4.2.2 The averaging method

The averaging method is to average multiple sets of data that are acquired under identical conditions except for motion [Wood and Henkelman 1986]. The artifacts in different data sets acquired at different times and therefore at different motion phases will be out of phase to each other and can cancel out each other. According to Wood and Henkelman [1986], the root-mean-squared error (RMSE) between the reference point image and the average point spread function (PSF) decreases as the number of averages increases. Often as many as 18 data sets are averaged in practice.

As with the gating methods, the problem of the averaging method is the increased data acquisition time.

1.4.2.3 The deliberate ghost positioning method

The deliberate ghost positioning method separates a part of ghosts away from the true position of the object by adjusting the pulse repetition time in order to avoid ghost overlapping the true position of the object [Wood and Henkelman 1986].

Note that it is not possible to eliminate ghosts by greatly increasing the separation y_G because any ghosts that should extend beyond field of view (FOV) are folded back into the image due to aliasing. However if the separation y_G is a multiple of the FOV, all the ghosts become superimposed on the true position of the object. In this case, the image is blurred worst. When the separation is a half-integral multiple of FOV, all the even-numbered ghosts and all the odd-numbered ghosts are superimposed separately. In this case it is possible to get better quality image if the FOV is big enough.

However the limitation of the deliberate ghost positioning method is that an exact synchronism between the motion and image acquisition is required, otherwise the method can be highly unpredictable.

1.4.2.4 The respiratory ordering of phase encoding (ROPE) method

The ROPE method alters the correspondence between physical time and the phase-encoding step k_y from linear relationship [Wood and Henkelman 1986]. The encoding steps are organized to make the reconstructed image appear as if only one monotonic movement were executed slowly. The reconstructed image becomes ghost free but remains blurred.

1.4.3 Fast Imaging Methods

Straightforwardly reducing data acquisition time can reduce the severity of motion effects on NMR signals and thus reduce artifacts caused by motions. For example, the imaging scan time of only a few seconds is involved in echo planar imaging (EPI) [Wehrli et al. 1988; Constable 1997]. However for these fast imaging methods, image contrast and resolution are often compromised [Wehrli et al. 1988; Constable 1997].

1.4.4 Post-processing Methods

The strategy of post-processing, or image processing, methods is to develop computer algorithms to remove motion artifacts mathematically from degraded data obtained by conventional imaging techniques. The advantage of post-processing methods is that no change to the conventional MRI system, nor to the imaging method, nor to the data acquisition procedure is needed. A detailed review of existing post-processing methods is given in Chapter 2.

As Wood and Henkelman [1986] pointed out, the spatially variant characteristics of motion, compounded with the ill-conditioned nature of image restoration, makes post-processing methods very complex and difficult mathematically. Most of existing post-processing methods work only for some very simple motions. However when applied to practical MRI images, these over-simplified, and thus non-realistic, motion models do not lead to adequate correction of artifacts.

The author spent some time examining the application of 3D filter techniques [Bruton and Bartley 1986; Bruton and Bartley 1985; Agathoklis and Bruton 1982] to remove motion artifacts from the k-space data. As this approach did not seem to be leading to a positive result, the author turned to investigating a new composite image processing method that was based on an empirical but more realistic motion model which takes into account the object's variant characteristics in both time and space. The non-rigidity of the imaged object was represented by a distortion transfer function (DTF). This model has been successfully applied to suppress ghosts caused by non-rigid, periodic motions along the slice selection axis for both phantom images and an clinical abdomen image.

1.5 Thesis Outline

This thesis is organized in the following way:

In Chapter 2, existing post-processing methods are reviewed in detail. It is shown that most of existing post-processing methods treated the motion in an over-simplified way. However when applied to practical MRI images, these over-simplified motion models often do not lead to adequate correction of artifacts.

In Chapter 3, we empirically propose a generalized motion model which does not treat motions as rigid as most existing post-processing methods do. The object's spatially variant characteristics are taken into account by introducing a distortion transfer function (DTF).

It is necessary for regions of support to be determined from MRI images in order to determine the DTFs and then correct the motion artifacts. In Chapter 4. a new contour detection technique: the snake, is studied in theory. Both continuous and discretized active contour models of snakes are introduced. In Chapter 5 the implementation and consideration of snakes is investigated. In Chapter 6, a contour-based region labeling method is proposed by the author. This new work allows segmentation of an image based on the snakes contour information.

In Chapter 7, we investigate how to estimate the distortion transfer functions given a corrupted image.

In Chapter 8, a composite image processing method to suppress ghosts caused by non-rigid, periodic motions along the slice selection axis is introduced. The strategy of this composite image processing method for ghost suppression is to make use of the estimated DTFs and the regions of support of the central image and ghosts. The correction results for both phantom images and the abdomen image with unknown respiratory motions are given.

Finally in Chapter 9 this thesis project is summarized and suggestions for future research are given.

CHAPTER 2

REVIEW OF POST-PROCESSING TECHNIQUES OF MOTION ARTIFACT SUPPRESSION IN MRI

2.1 Introduction

In this chapter, existing post-processing methods for motion artifact suppression in MRI are reviewed. A model of translational motion in image plane is mathematically described. Two methods for suppressing artifacts caused by motions of this model are introduced. Then a model of planar rotational motion of a non-rigid body and the corresponding artifact suppression method are presented. A model involving amplitude modulation (AM) of the k-space data is used to characterize the effect of periodic motions along the slice selection axis. Finally post-processing work for motion artifact suppression from a number of other research groups is introduced.

2.2 The Rigid Translational Motion in the Imaging Plane

Many papers have assumptions involving planar rigid translational motions in the x and y directions [Korin et al. 1989; Hedley et al. 1991b; Zoroofi et al. 1996; Ehman and Felmlee 1989; Korin et al. 1990]. The following constraints are typically émployed:

- The motion is rigid. In other words, every point of the object moves identically.
- The motion is translational, i.e. it involves just simple displacements along the x and y directions.
- The motion is only in the imaging plane with the object remaining in the field of view (FOV).

Let's analyze effects of such a motion model on the received k-space data. Assume the displacement of the object due to translational motion is $p(t)$ and $q(t)$ in the

x and y directions respectively. The time t is assumed be zero at the beginning of scanning. The initial displacements are set as: $p(0) = q(0) = 0$. Then the corrupted signal is given by:

$$s_{motion}(k_x, k_y) \propto \int\int \rho(x - p(t)), y - q(t)) \exp(-j2\pi(k_x x + k_y y))dxdy \qquad (2.1)$$

As discussed in Chapter 1, the read-out time for a phase-encoding step is very short and much less than the repetition time T_R between two consecutive phase-encoding steps. This implies that intraview motion (during the read-out time) can be ignored when interview motion (between phase-encoding steps) is present. The relationship between time t and phase-encoding step k_y is linear when the phase-encoding gradient is monotanically incremented with time. Thus the motion is actually a function of k_y only. That is $p(t) = p(k_y)$, $q(t) = q(k_y)$. Using the Fourier relationship between position shift in space domain and phase in frequency domain, we can write the right hand side of Equation 2.1 as:

$$\int\int \rho(x - p(k_y), y - q(k_y)) \exp(-j2\pi(k_x + k_y))dxdy =$$

$$\int\int \rho(x, y) \exp(-j2\pi p(k_y)x) \exp(-j2\pi q(k_y)y) \exp(-j2\pi(k_x x + k_y y))dxdy$$

As we have seen in Chapter 1 in Equation 1.1, $s_{no-motion}(k_x, k_y)$, the frequency data when there is no motion, is given by:

$$s_{no-motion} \propto \int\int \rho(x, y) \exp(-j2\pi(k_x x + k_y y))dxdy$$

so that

$$s_{motion}(k_x, k_y) \propto \exp^{-j\phi(k_x,k_y)} s_{no-motion}(k_x, k_y) \qquad (2.2)$$

and

$$\phi(k_x, k_y) = 2\pi[k_x p(k_y) + k_y q(k_y)] \tag{2.3}$$

Equation 2.2 shows that the difference between $s_{motion}(k_x, k_y)$ and $s_{no-motion}(k_x, k_y)$ is $\exp(-j\phi(k_x, k_y))$. This implies that the planar rigid translational motion only introduces a phase shift to the uncorrupted frequency data $s_{no-motion}(k_x, k_y)$ without altering its magnitude distribution.

To visualize artifacts from such a translational motion, motion corrupted phantom images can be simulated. To avoid introducing data anomalies [Smith et al. 1997] the phantom data should be generated directly in k-space rather than Fourier transforming the discretely sampled image data. The k-space technique avoids producing k-space data with improper digital signal processing characteristics and produces k-space data equivalent to that obtained actually imaging an object in a real MRI system. This k-space technique for valid phantom generation will be discussed in detail in a future paper by Smith and Yang [1998].

Fig. 2.1 shows the directly calculated frequency data (upper) and the corresponding reconstructed images (lower) for two phantoms when there exists no motion. The image sizes of these two phantoms and all other phantoms in this thesis are 256 by 256. The simple phantom image (left) will allow us to see more clearly the basic effects of the artifacts. This simple phantom image consists of two ellipsoids, whose parameters are listed in Table 2.1 assuming the image has (0,0) in the centre and extends from -1 to 1 along the x and y directions. In the table, coordinates of the centre of an ellipsoid in the x and y directions are denoted as X-centre and Y-centre respectively. The radii in the x and y directions are represented by X-radius and Y-radius respectively. The angle between the major axis of an ellipsoid and the x coordinate axis is denoted as Angle. The strength of an ellipsoid is represented by Intensity. The other phantom on the right is a more complicated Shepp-Logan phantom variant, which is designed

No.	X-centre	Y-centre	X-radius	Y-radius	Angle	Intensity	Description
1	0.0	0.0	0.48	0.30	0.0	-5.004	the inner boundary
2	0.0	0.0	0.53	0.40	0.0	5.004	the outer boundary

Table 2.1. Centre position, radius size, angle and intensity parameters of ellipsoids in the simple phantom image

to mimic many of the characteristics of the human abdomen [Smith and Yang 1998]. The smaller objects are intended to represent lesions or other image features easily obscured by motion distortion. The parameters about the ellipsoids associated with forming this phantom are listed in table 2.2.

The k-space characteristics of an object with inter-view motion can be simulated by using the conceptual framework suggested by Xiang and Henkelman [1993] discussed in Chapter 1. According to [Xiang and Henkelman 1993], each measured row of k-space data set corresponds to a line of the Fourier transform of the imaged object selected by $z = z_0$ at a particular phase-encoding time.

In case of rigid planar translational motion, where the imaging plane $z = z_0$ changes as $\rho(x - p(k_y), y - q(k_y))$, the Fourier transform of the imaging plane at the m^{th} phase-encoding step is:

$$F(k_x, k_y) = \mathcal{F}(\rho(x - p(m), y - q(m)))$$

Thus the k-space data row corresponding to this phase-encoding step m is given by:

$$S_{motion}(n, m) = F(k_x, k_y)|_{k_x=n, k_y=m}, \qquad n = 0, 1, ..., N_x - 1$$

where N_x is the length of the recorded NMR signal.

For the phantom image when only inter-view motion is considered, the k-space data set can be generated row by row in the frequency domain directly by the following pseudo code as:

```
for(ellipsoid l from 1 to L)
```

```
    for(phase-encoding step $k_y$ from 0 to $N_y - 1$)
        for(read-out step n from 0 to $N_x - 1$)
            $F(k_x, k_y) = \mathcal{F}(\rho_l(x - p(m), y - q(m))$
            $S_{l_{motion}}(n, m) = F(k_x, k_y)|_{k_x=n, k_y=m}$
        endfor
    endfor
endfor
The motion corrupted phantom data $S_{motion}(n, m) = \sum_{l=1}^{L} S_{l_{motion}}(n, m)$
```

By using this approach, the two motion corrupted phantom images as shown in Fig. 2.2 were generated with shiftment only along the y direction by setting $p(k_y) = 0$ and $q(k_y) = 0.05 \cos(8.0 \cdot 2\pi k_y / N_y)$.

The artifacts display both blurring and ghosts in Fig. 2.2. In our experiments, we have seen that the distances between ghosts became bigger when the number of motion cycles increased in agreement with the work of Wood and Henkelman [1985].

The spectrum shift method and iterative phase retrieval method are common artifact suppression methods based on the planar rigid translational motion assumption.

2.2.1 The spectrum shift method

According to the spectrum shift method, the correction of the phase shift is done by just applying an opposite phase factor $\phi(k_x, k_y)$ to the motion corrupted data.

1. Correction when motion parameters are known

Korin *et al.* [1989] studied a very simple case: the motion was assumed to be linear in time with a known velocity. The k-space data was recorded by placing the phantom material on a table and translating the table through the gantry at an operator-controlled speed. As the motion parameters are known, the phase factor $\phi(k_x, k_y)$ is easily calculated according to Equation 2.3. Many data sets from the experimental system with different motion parameters, such as different velocities main organs

No.	X-centre	Y-centre	X-radius	Y-radius	Angle	Intensity	description
1	0.0	0.0	0.69	0.54	0.0	6.004	outer wall
2	-0.01144	-0.01677	0.63	0.51	0.0	-1.987	inner wall
3	0.0	0.0	0.58	0.45	0.0	2.45	organ 1
4	0.0	0.47	0.029	0.025	0.0	5.019	organ 2
5	0.005	0.325	0.17	0.12	0.0	2.45	organ 3
6	-0.24	0.35	0.091	0.05	25.0	2.01	organ 4
7	0.24	0.353	0.080	0.05	155.0	2.01	organ 5
8	-0.28	0.0018	0.286	0.286	28.0	2.43	organ 6
9	-0.1	0.04	0.1	0.07	110.0	0.5	outer wall of organ 7
10	-0.1	0.04	0.08	0.05	110.0	-0.2	organ 7
11	0.3	0.008	0.29	0.288	15.0	2.43	organ 8
12	0.35	0.0	0.140	0.09	40.0	-0.8	wall of organ 9
13	0.34	0.001	0.11	0.076	40.0	0.9	organ 9
14	0.05	-0.35	0.058	0.07	0.0	2.0	wall of organ 10
15	0.05	-0.35	0.055	0.06	0.0	-1.6	organ 10
16	0.05	-0.325	0.115	0.125	0.0	1.31	wall of organ 11
17	0.05	-0.33	0.1	0.12	0.0	-1.8	organ 11

highly detailed left side

No.	X-centre	Y-centre	X-radius	Y-radius	Angle	Intensity	description
18	-0.3	0.4	0.01	0.008	0.0	3.01	detail 1
19	-0.6	0.04	0.008	0.009	30.0	4.01	detail 2
20	-0.62	-0.001	0.01	0.01	0.0	5.01	detail 3
21	-0.6	-0.07	0.01	0.02	0.0	4.01	detail 4
22	-0.38	-0.38	0.015	0.009	0.0	4.01	detail 5
23	-0.49	0.001	0.001	0.007	0.0	2.09	detail 6
24	-0.45	0.14	0.05	0.035	0.0	1.019	structure 1
25	-0.45	0.07	0.052	0.035	0.0	1.019	structure 2
26	-0.45	-0.0	0.0521	0.036	0.0	1.019	structure 3

low detailed right side

No.	X-centre	Y-centre	X-radius	Y-radius	Angle	Intensity	description
27	0.25	-0.45	0.008	0.009	0.0	2.01	detail 7
28	0.54	-0.20	0.01	0.008	50.0	5.01	detail 8
29	0.48	0.001	0.009	0.009	0.0	1.51	detail 9
30	0.20	-0.20	0.009	0.012	20.0	1.01	detail 10

lesions

No.	X-centre	Y-centre	X-radius	Y-radius	Angle	Intensity	description
31	-0.3	0.17	0.03	0.06	20.0	1.019	lesion 1
32	-0.3	-0.20	0.05	0.02	0.0	1.01	lesion 2
33	-0.2	0.15	0.01	0.008	0.0	1.01	lesion 3
34	0.01	0.34	0.03	0.05	0.0	0.41	lesion 4
35	0.002	0.16	0.039	0.039	0.0	2.01	lesion 5
36	0.006	-0.16	0.043	0.043	0.0	2.01	lesion 6
37	0.3	0.2	0.07	0.04	120.0	0.5	lesion 7
38	0.42	-0.15	0.03	0.05	35.0	0.9	lesion 8

Table 2.2. Centre position, radius size, angle and intensity parameters of the ellipsoids in the abdomen phantom image

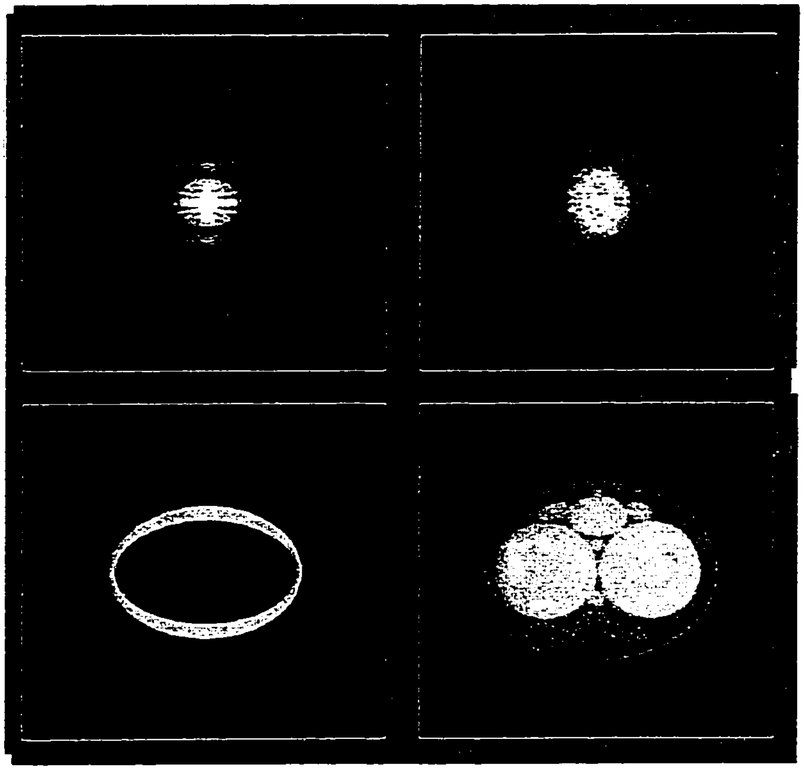


Figure 2.1. The k-space data (upper) and the reconstructed images (lower) for two phantoms without motion artifacts (left: simple phantom, right: abdomen phantom)

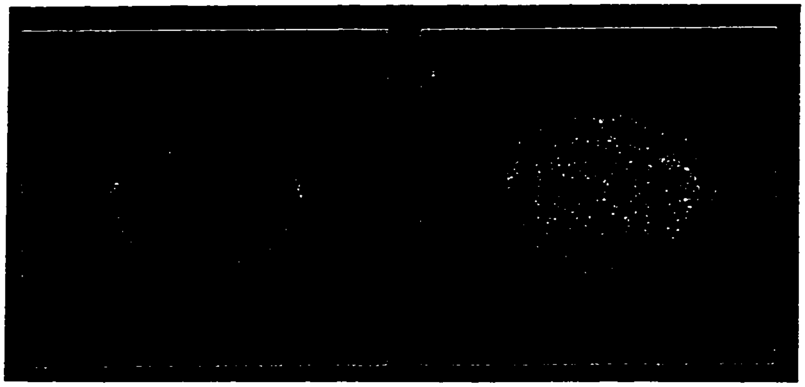


Figure 2.2. Phantom images with motion corruption simulated by the planar rigid translational motion model

and directions (or angles), were tested. The experiment results agreed that this rigid translational motion in the imaging plane only introduces a phase shift in k-space data.

2. Correction when motion parameters are unknown

Based on the same planar rigid translational motion assumption, Ehman and Felmlee [1989] investigated a more realistic case: the motion was arbitrary in time and the motion parameters were not identical during the data acquisition time. Of course the problem is more complicated than the above one with motion parameters known. Ehman and Felmlee [1989] treated the motion as piecewise linear in time. The displacements were different for different views and the phase shifts changed correspondingly. The correction needed to be adaptive.

According to Ehman and Felmlee [1989] the displacement that took place between different views was measured by using specially encoded "navigator" echoes that were interleaved with the image sequence. The navigator echo sequence designed to provide x direction displacement was termed as x-NAV. The NAV echo was similar to an image echo, except that no phase-encoding was applied. This implies that x-NAV echo data varies from view to view only if motion along the x axis is present. The process is handled by doing an one-dimension Fourier transform (1D FT) of the received navigator echoes and calculating the cross correlation between that of the current navigator echo and that of a reference echo. The displacement between the two phase steps is treated equal to the lag of the maximum value of the cross correlation. Using this approach the motion parameters are measured.

Similarly the y-NAV, xy-NAV and z-NAV can be employed to measure the displacements in the y direction, xy plane and z direction respectively. After the displacements of all views are measured by using the navigator echoes, the correction can be done by the spectrum shift method. Based on the same planar rigid translational motion assumption, Korin *et al.* [1990] applied that method to three-dimensional

MRI. Felmlee *et al.* [1991] developed a method to adaptively compensate for motion artifacts without use of navigator echoes. They used the phase-encoding image data itself, rather than the separated non-phase-encoded navigator echoes, to evaluate the interview displacement along the read-out axis by using the so called edge-detection algorithm. The process involves first 1D Fourier transforming the measured k-space data along the x axis to get the hybrid space data set $l_j(x)$, where the index j indicates the phase-encoding step. Then the magnitude data $|l_j(x)|$ are compared to a threshold progressively from the outer edge of the field toward the center. The object "edge" is assigned to the spatial location where the data value exceeds the threshold. The thresholds are changed empirically along the k_y direction on the basis of the expected signal density. For experiments, the motions are controlled to be planar rigid translational. The focus of these post-processing methods are how to determine motions' parameters at each phase-encoding step.

2.2.2 The iterative phase retrieval method

Still based on the same planar rigid translational motion assumption. Hedley *et al.* [1991b] proposed the so called generalized projection method to suppress artifacts with motion parameters unknown. Unlike Ehman and Felmlee [1989], the generalized projection method does not need to estimate motion parameters. As the translational motion introduces only a phase error, Hedley *et al.* [1991b] employed the Gerchberg Saxton (GS) method, a general phase correction method. During each iteration of the GS algorithm, a constraint in frequency domain was imposed to the corrected k-space data. The constraint Hedley used was that the magnitude of the corrected frequency data should be the same as that of the measured k-space data. The corrected k-space data after applied this constraint was inverse Fourier transformed to the space domain. In the spatial domain another constraint was imposed. The image value were set to zero outside the ROS of the object. Then the spatial data was Fourier transformed back to the frequency domain to start the next iteration.

The GS method was demonstrated by Hedley's computer simulations to be effective to remove the artifacts for phantom images.

Later, Hedley *et al.* [1991a] reported further work on this GS method. This time they aimed at improving convergence properties by incorporating additional *a priori* information specific to their situation. The incorporated *a priori* information they used is that the phase shift should have a linear relationship to k_y when this phase shift is in the range of $(-\pi, +\pi)$ and not wrapped. They used a least square estimate (LSE) to fit a line to the phase shift in the centre of the k-space data. The phantom with simulated motion in the imaging plane showed that the new algorithm did improve the convergence rate.

Zoroofi *et al.* [1995] improved the phase retrieval method by using two successive steps: first the spectrum shift method was applied to remove artifacts caused by the x direction motion. Then the phase retrieval method was employed to eliminate the remaining artifacts caused by subpixel motion in both the x and y directions. The correction results for simulated motions and experimentally controlled motions were successful.

2.3 Rotational Motion of a Rigid Body

Zoroofi *et al.* [1996] studied the motion using a rigid translational and rotational model. As we already have shown, the rigid translational motion introduces a phase error and does not alter the amplitude distribution. On the other hand, according to Fourier theory, rotation of a rigid body about a fixed point in the spatial domain introduces an identical rotation in the corresponding frequency domain as:

$$S_{rotation}(k_x, k_y) = S_{no-motion}(k_x \cos\theta_r(k_y) + k_y \sin\theta_r(k_y),\ \ -k_x \sin\theta_r(k_y) + k_y \cos\theta_r(k_y)) \tag{2.4}$$

where θ_r is the rotational angle, $s_{no-motion}(k_x, k_y)$ is the true image without motion and $S_{rotation}(k_x, k_y)$ is the k-space data corrupted by the rotational motion.

Equation 2.4 indicates that the rotational motion about a fixed point in the imaging plane corrupts the MR image with nonuniform sampling in frequency domain.

Fig. 2.3 shows the two phantom images corrupted by rigid planar rotational motion by using Xiang's conceptual approach (see previous section). The rotation angle was designed as: $\theta_r(k_y) = 0.005\cos(8.0 \cdot 2\pi k_y/N_y)$.

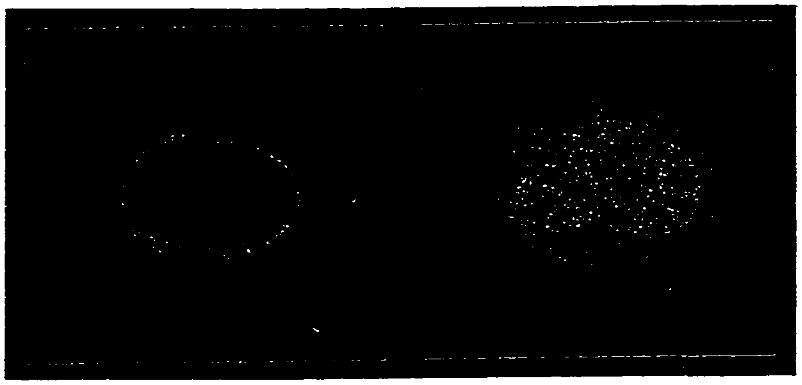


Figure 2.3. Phantom images with motion corruption simulated by the rotational motion model

If both the translational motion and the rotational motion are considered. then the corrupted k-space data $s_{motion}(k_x, k_y)$ can be described by:

$$S_{motion}(k_x, k_y) = \exp(-j2\pi(p(k_y)k_x + q(k_y)k_y)) \cdot S_{rotation}(k_x, k_y)$$

This implies the general planar rigid motion (translational and rotational) of the object imposes both a phase error and nonuniform sampling on the MRI signal. The phase error can be easily removed by applying the spectrum shift method or the phase retrieval method discussed in last section. It was suggested that the nonuniform sampling be removed by interpolation methods [Zoroofi et al. 1996; Atalar and Onural 1992]. Zoroofi *et al.* [1996] also tried to estimate the unknown motion parameters from the corrupted k-space data by minimizing the energy outside the boundary of the imaged object. These methods were applied to mathematical phantoms and demonstrated efficient.

2.4 The Motion along the Slice Selection Axis

Mitsa *et al.* [1990] published their work about rigid, periodic motion along the slice selection axis. The motion was modeled as an amplitude modulation (AM) of the true MRI data without motion. The model, which is also referred to as Mitsa's AM model, was derived from the physical background about movement of a point source. The amplitude modulation factor, representing the motion's periodicity in time, is chosen to be an expansion of Fourier series as:

$$s_{motion}(k_x, k_y) = s_{no-motion}(k_x, k_y)[1 + \sum_{-\infty}^{+\infty} \frac{\Delta m_n}{m_0} \cos(\frac{2\pi n N_b k_y}{N_y \Delta G} + \phi_n)] \qquad (2.5)$$

where m_0 is the intrinsic strength of a stationary point source, Δm_n represents the change in strength due to the n^{th} harmonic of the periodic motion. N_b is the number of movement cycles during the entire data acquisition. ΔG is the step used to increment k_y. N_y is the number of phase encoding steps. ϕ_n refers to the phase of periodic motion with respect to the start of data acquisition for the n^{th} harmonic.

Fig. 2.4 shows the corrupted phantom images of Fig. 2.1 simulated by using Equation 2.5. Instead of an infinite Fourier series terms, only the first three terms are used for generating Fig. 2.4:

$$1 + 0.5\sin(\tfrac{2\pi k_y}{4.8N_y} + 0.785) + 0.15\sin(\tfrac{2\pi k_y}{2.4N_y} + 1.57) + 0.005\sin(\tfrac{2\pi k_y}{1.2N_y} + 3.141)$$

In Fig. 2.4, ghosts are much more severe than blurring and look like scaled and shifted versions of the true image of the object.

To identify and then correct for the amplitude modulation factor, the effect of amplitude modulation factor is emphasized by projecting the k-space data along the read-out direction. This 1D projection displays peaks in its power spectrum. The smoothed projection was gotten by using an appropriate notch filter and considered

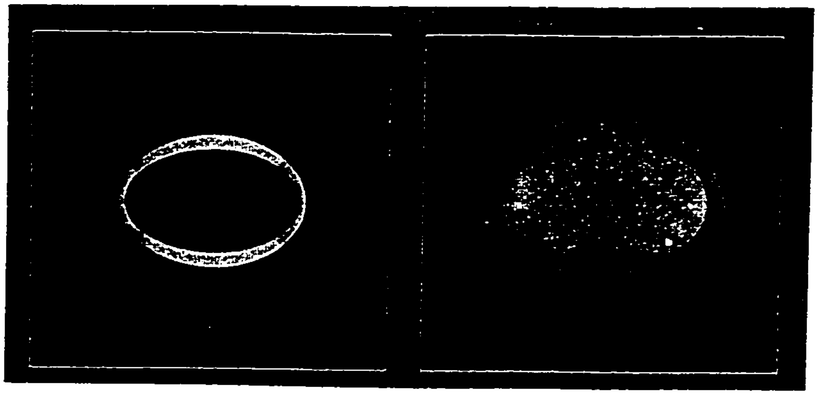


Figure 2.4. Z plane motion corrupted phantom images simulated by Mitsa's AM model

as the 1D projection of the true data without motion. Then the motion kernel was estimated as the ratio of the raw projection to the smoothed projection. The Mitsa's post-processing algorithm consists of the following detailed steps:

1. Project the magnitude of the measured k-space data along the x direction to form one line of projection data with respect to position in k_y, the phase-encoding direction. The central 31 columns of data associated with DC component and low frequency components are removed when projecting the measured k-space data in order to obtain sharper peaks and a better scaling scheme [Zeng 1996].

2. Perform an inverse discrete Fourier transform (DFT) to the projected data to obtain the power spectrum of the projected data.

3. Identify the energy peaks in the power spectrum related to the motion.

4. Apply a notch filter to the power spectrum to eliminate the identified motion peaks.

5. Fourier transform the peak removed power spectrum to obtain an estimate of the projection data without motion.

6. The ratio of the projected data from the measured k-space data to the projection data with the motion peaks removed is considered as an estimate of the motion kernel corresponding to the amplitude modulation factor in the motion model.

7. Divide each row of the measured k-space data by the motion kernel and get the corrected k-space data.

8. Reconstruct the image by applying the 2D DFT to the corrected k-space data. This image is the corrected image with motion artifacts suppressed.

Hedley and Yan [1992] furthered Mitsa's work. Their motion model is still the amplitude modulation model proposed by Mitsa. Instead of using Mitsa's projection method, Hedley presented an alternative method of using linear equations to obtain the motion kernel by minimizing data outside a manually determined region of support (ROS) of the true image.

2.5 A Review of Post-Processing Algorithms for Special Motions

2.5.1 The Discontinuous Motion

Wood *et al.* [1995] studied a special case of head imaging in which the volunteer subject remained stationary except for nodding his head once during the data acquisition time. Because the motion occurs only during a part of the data acquisition time and there is no motion during the remainder of the data acquisition time, we call this discontinuous motion.

Wood *et al.* [1995] pointed out that the movements caused abrupt changes at the interface between views along the k_y direction. By using the *t-test* method, the k-space data can be separated automatically into several segments with some of the segments without motion, some of the segments with different motion parameters. Then each k-space segment is Fourier transformed into the image domain in which rotational angles and translational displacements are measured manually. Then segments with different motion parameters are corrected separately with their own motion parameters.

2.5.2 The Linear Expansion Motion

Atalar and Onural [1992] modeled respiratory motions as a combination of rigid translational and linear expansion motions in the imaging plane. As discussed earlier, the rigid translational motion only introduces a phase error to the true data without motion. The spectrum shift method or the phase retrieval method can be used to compensate the phase error. On the other hand. in the linear expansion motion. there is no motion at the centre (origin) and the amplitude of the motion is proportional to the distance from the center. The linear expansion motion changes the rectangular samples into nonrectangular samples. The correction of the linear expansion motion requires recovery of the rectangular samples from the nonrectangular samples. Atalar and Onural [1992] suggested applying interpolation methods. They pointed out that the positions of the nonrectangular samples were not arbitrary. The samples are taken on a line but at a different position and the samples on this line are taken nonuniformly. Thus the recovery of the rectangular samples from the nonrectangular samples is reduced to two one-dimensional interpolations.

Atalar and Onural [1992] investigated five interpolation methods. The simulation results demonstrated the efficiency of the methods for mathematical phantom images corrupted by a combination of rigid translational and linear expansion motion.

2.5.3 Riek's POCS Method for Multislice MR Images

Riek *et al.* [1995. 1994] investigated the suppression of artifacts caused by out-of-plane motions in multislice MRI by using a non-linear method called POCS (projection onto convex sets). Four constraints were used [Riek et al. 1995]:

- The reconstructed images were real.
- The real parts of the reconstructed images had nonnegative values.
- Image values were zero outside the region of support.
- Errors between the measured and modeled data should be within a confidence bound.

2.6 Work in Our Research Group

J. Zeng, a graduate student in Dr. Smith's group from 1994-1996, devoted her M.Sc. thesis project to motion artifact suppression by using post-processing techniques [Zeng 1996]. She recorded an abdominal image as shown in Fig. 2.5 by using the clinical MRI system in Foothill's Hospital in Calgary.

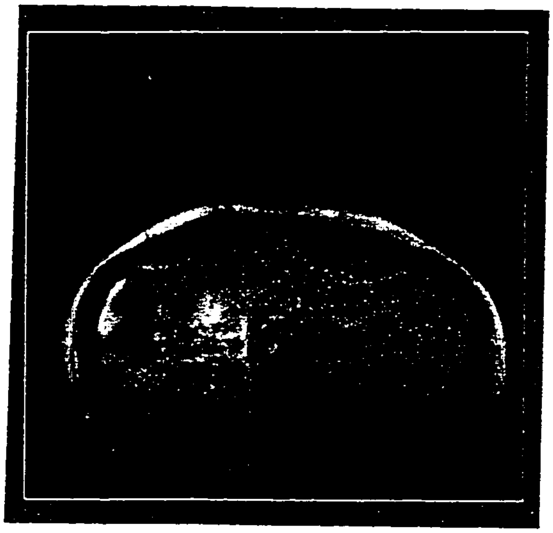


Figure 2.5. An abdomen image from an MRI system showing motion artifact with unknown characteristics

The abdomen image was corrupted primarily by unknown respiratory motion. The artifacts displayed in the abdomen image appeared to correspond to the ghosts displayed in the corrupted phantom image simulated by Mitsa's AM motion model shown in Fig. 2.4. Thus Zeng focused on Mitsa's AM model and made a number of improvements to Mitsa's approach:

- A projection power spectrum with higher resolution than by using DFT was achieved by using an ARMA model [Smith et al. 1986]. Thus the accuracy of peak identification, and associated artifact suppression could be expected to be improved.
- The Blackman and Harris (BH) window was introduced to make the notch fil-

ter smooth. This avoided introducing distortion because of the filtering action [Harris 1978].

- To automatically detect the peaks. the adaptive algorithm RLS was employed [Zeng 1996].

However, despite these improvements to Mitsa's method and many successes with phantom images. Zeng reported being unable to suppress the motion artifacts for the abdomen image. She stated in her thesis: Mitsa's motion model is too simple to describe correctly the actual abdominal motion [Zeng 1996].

2.7 Summary

To mathematically remove motion artifacts by post-processing algorithms in MRI is very attractive. The work in this area began from early 1980s. Experimental and analytical investigations showed that motions during MRI data acquisition cause artifacts to the reconstructed MRI images. The artifacts could display as blurring and/or ghosts. Many existing post-processing methods of motion artifact suppression are based on some simplified assumptions such as rigid translational motions in the imaging plane. rotational motion of a rigid body in the imaging plane and rigid. periodic motions along the slice selection axis. The artifacts displayed in the abdomen image recorded from a real MRI system seemed similar to ghost artifacts generated by using Mitsa's AM motion model. However the correction results obtained by using Mitsa's approach were disappointing. The problem was pointed out as lack of an appropriate model for real motions. In the remainder of this thesis we shall discuss an empirical motion model to better reflect the practical situation.

CHAPTER 3

A GENERALIZED MOTION MODEL

3.1 Introduction

As reviewed in Chapter 3, the existing post-processing models for motion artifact suppression are not realistic. Many of the artifact suppression techniques are based on some over-simplified assumptions about motions and thus often fail to work. It is a challenge to propose more realistic motion models and develop corresponding algorithms that work for not only phantom images but also for real MR images.

In this chapter a general empirical motion model is proposed. The key point of this generalized motion model is to introduce a distortion transfer function (DTF). a factor that represents the spatially variant characteristics of the motion or non-rigidity of the imaged object. Then we show that the existing post-processing approaches discussed in Chapter 2 are just special cases of the proposed generalized model. Finally phantom images corrupted by a variety of DTFs are generated to show the different characteristics of such ghosts.

3.2 The Generalized Motion Model

Let's examine the abdomen image in Fig. 2.5. It seems true that the displayed ghosts are very similar to the ghosts simulated by Mitsa's AM model for periodic motion as shown in Fig. 2.4. That is natural considering the physical background that the respiratory motion suffered by the abdomen is a kind of periodic motion along the slice selection axis. However if you observe carefully, you can see that the ghosts displayed in the abdomen image are not simply scaled and shifted versions of the true object image as suggested by Mitsa's AM model. We attribute the difference between the actual image and Mitsa's theoretical model to the fact that Mitsa's assumption that every point in the image subject moves identically is not appropriate.

Mitsa's treatment of the motion as rigid is not realistic. We would like to present a new generalized motion model capable of handling motion of non-rigid objects. The empirical theory behind this generalized motion model is discussed below.

According to Equation 1.1, the k-space data without motion is:

$$s_{no-motion}(k_x, k_y) \propto \int_y \int_x \rho(x, y) \cdot \exp(-j2\pi x k_x) \cdot \exp(-j2\pi y k_y) dx dy \tag{3.1}$$

We empirically model motions that occur during the data acquisition as:

$$\begin{aligned} s_{motion}(k_x, k_y) \quad \propto & \\ & \int_y \int_x \rho(f_x(x, y, k_x, k_y), f_y(x, y, k_x, k_y)) \exp(-j2\pi x k_x) \cdot \exp(-j2\pi y k_y) \\ & \cdot F_{out-of-plane}(k_x, k_y) \cdot F_{non-rigid}(k_x, k_y) dx dy \end{aligned} \tag{3.2}$$

where $f_x(x, y, k_x, k_y)$ and $f_y(x, y, k_x, k_y)$ represent rigid translational motions in the imaging plane's x and y directions respectively. $F_{out-of-plane}(k_x, k_y)$ characterizes rigid motions out of the imaging plane. The possible non-rigid motion is modeled by introducing another factor $F_{non-rigid}(k_x, k_y)$, also called the Distortion Transfer Function (DTF). The inverse Fourier transform of $F_{non-rigid}(k_x, k_y)$ represents how a point in the image domain is distorted because of the non-rigidity of the object in motion.

When intraview motions are ignored, view-to-view motions make the various modulating terms related to motion in time become function of k_y only as:

$$\begin{aligned} f_x(x, y, k_x, k_y) = f_x(x, y, k_y), \quad f_y(x, y, k_x, k_y) = f_y(x, y, k_y) \\ F_{out-of-plane}(k_x, k_y) = F_{out-of-plane}(k_y) \end{aligned} \tag{3.3}$$

Motion models used by the existing post-processing approaches discussed in Chapter 2 are just special cases of Equation 3.2 as the following shows:

- For general rigid motion in the imaging plane

$$F_{non-rigid}(k_x, k_y) = 1, \quad F_{out-of-plane}(k_x, k_y) = 1$$

- For rigid translational motions in the imaging plane [Korin et al. 1989; Hedley et al. 1991b; Zoroofi et al. 1996; Ehman and Felmlee 1989; Korin et al. 1990]

  $$F_{non-rigid}(k_x, k_y) = 1, \quad F_{out-of-plane}(k_x, k_y) = 1$$

  $$f_x(x, y, k_x, k_y) = x - p(k_y), \quad f_y(x, y, k_x, k_y) = y - q(k_y)$$

  so that

  $$s_{motion}(k_x, k_y) = s_{no-motion}(k_x, k_y) \cdot \exp(-j2\pi[k_x p(k_y) + k_y q(k_y)])$$

  where functions $p(k_y)$ and $q(k_y)$ describe displacements in the x and y direction respectively. The same conclusion as in Chapter 2 is obtained: only a phase distortion is introduced to the k-space data in the case of rigid translational view-to-view motion in the imaging plane.

- For rigid motions along the slice selection axis

  $$F_{non-rigid}(k_x, k_y) = 1$$

  $$f_x(x, y, k_x, k_y) = x, \quad f_y(x, y, k_x, k_y) = y$$

  so that

  $$s_{motion}(k_x, k_y) = s_{no-motion}(k_x, k_y) \cdot F_{out-of-plane}(k_y)$$

  In addition, if the motion along the slice selection axis is periodic, then the function $F_{out-of-plane}(k_y)$ becomes a periodic function as suggested by Mitsa *et al.* [1990].

3.3 The Generalized Motion Model for Non-Rigid Periodic Motions along Z Axis

Our mathematical model Equation 3.2 can be applied generally. However, our discussion with the abdomen as the studied object is confined to the reconstruction of images corrupted by non-rigid periodic motions along the z axis.

As we discussed earlier, artifacts caused by periodic motions along z-axis are characterized as ghosts. By the generalized motion model, ghosts from non-rigid body motion are modeled as distorted replicas of the true image without motion. These distorted replicas are shifted from and often overlapped with the true or central image and each other. The displacements of ghosts from the central image are determined by the number of cycles of the motion that occurred during the data measurement [Wood and Henkelman 1986]. Therefore, the Fourier transform of the n^{th} ghost image becomes:

$$s_{ghost_n}(k_x, k_y) = s_{no-motion}(k_x, k_y) \cdot DTF_n(k_x, k_y) \cdot \exp(-j2\pi\Delta_n k_y) \qquad (3.4)$$

where the Δ_n term is the k-space data's phase shift related to the relative position of the ghost and central image. The motion is fully described by the combination of this phase term, which represents the motion's characteristics in time, and a distortion transfer function (DTF) term, which expresses the motion's spatially variant characteristics. Comparing Equation 3.2 and Equation 3.4, we can consider $F_{out-of-plane}(k_y) = \exp(-j2\pi\Delta_n k_y)$ and $F_{non-rigid}(k_x, k_y) = DTF_n(k_x, k_y)$.

As the corrupted image is composed of the true image and all the ghosts, possibly overlapping in image domain, we have:

$$s_{motion}(k_x, k_y) = s_{no-motion}(k_x, k_y) + s_{ghost_1}(k_x, k_y) + \ldots + s_{ghost_N}(k_x, k_y)$$

which reduces to

$$s_{motion}(k_x, k_y) = s_{no-motion}(k_x, k_y)(1 + \sum_{n=1}^{N} DTF_n(k_x, k_y)\exp(-j2\pi\Delta_n k_y)) \qquad (3.5)$$

where N is the number of ghosts displayed in the corrupted image. For every ghost the displacement Δ_n and spatially variant characteristics $DTF_n(k_x, k_y)$ may be different.

Mitsa's model [Mitsa et al. 1990] is just a special case of our generalized motion model expressed by Equation 3.5. His ghosts appear as pairs with their $DTF_n(k_x, k_y)$ identical and equal to the same strength constant, and with their displacements Δ_n equal in magnitude and opposite to each other.

Equation 3.5 indicates that there are two parts in the corrupted k-space data: the k-space data of the true image, $s_{no-motion}(k_x, k_y)$, and the k-space data of all the ghosts, $s_{ghost}(k_x, k_y)$. The two parts are related as:

$$s_{ghost}(k_x, k_y) = s_{no-motion}(k_x, k_y) \cdot \sum_{n=1}^{N} DTF_n(k_x, k_y)\exp(-j2\pi\Delta_n k_y) \qquad (3.6)$$

3.4 Examples of Distortion Transfer Functions

To visualize the ghosts caused by non-rigid periodic motions along z-axis, eight different distortion transfer functions (DTFs) were generated to simulate motion corruptions. The DTFs were modeled after some simple filter characteristics.

The first DTF was chosen to have the form of a Gaussian low-pass filter to introduce some simple motion blurring into the ghost images. The mathematical description of this DTF is as:

$$F_{non-rigid}(k_x, k_y) = \frac{\alpha}{\sqrt{2\pi}\sigma}\exp(-D^2(k_x, k_y)/\sigma^2) \qquad (3.7)$$

where α is a scaling coefficient. $D(k_x, k_y)$, equal to $\sqrt{k_x^2 + k_y^2}$, is the distance to the origin point in frequency domain coordinate system. σ is the parameter for controlling width of the filter. The frequency domain image of this DTF with $\alpha = 0.6, \sigma = 25.0$

is shown in Fig. 3.1(left). The image size is 256 by 256. We chose the basic motion parameters as $N = 2$, $\Delta_1 = -80$. and $\Delta_2 = 80$ to mimic the number and positions of the ghosts in the abdomen image in Fig. 2.5. By putting these parameters into Equation 3.5, the motion corrupted phantom image corresponding to that shown in Fig. 2.1(right) is simulated as shown in Fig. 3.1(right).

Figure 3.1. A DTF having a form of Gaussian low-pass filter (left) and the corresponding motion corrupted phantom image (right)

Then a DTF having the characteristics of a high-pass filter was chosen to produce ghosts that have the form of outlines of the central image. The abdomen image appears to have ghosts whose high frequency edges are enhanced. Mathematically the behavior of high-pass filters in passing or suppressing frequencies is exactly the opposite to that of low-pass filters. The transfer function of a high-pass filter can be obtained simply by "flipping" that of a low-pass filter, that is:

$$F_{non-rigid}(k_x, k_y) = \alpha \cdot (1 - \frac{1}{\sqrt{2\pi}\sigma} \exp(-D^2(k_x, k_y)/\sigma^2)) \tag{3.8}$$

Fig. 3.2(left) shows the DTF having a form of Gaussian high-pass filter with the filter characteristics chosen as $\alpha = 0.6$, $\sigma = 25.0$. The corresponding corrupted phantom image is shown in Fig. 3.2(right).

Comparing the two motion corrupted phantom images in Fig. 3.1 and Fig. 3.2, we can see clearly the difference in ghost appearance. A DTF having a low-pass filter

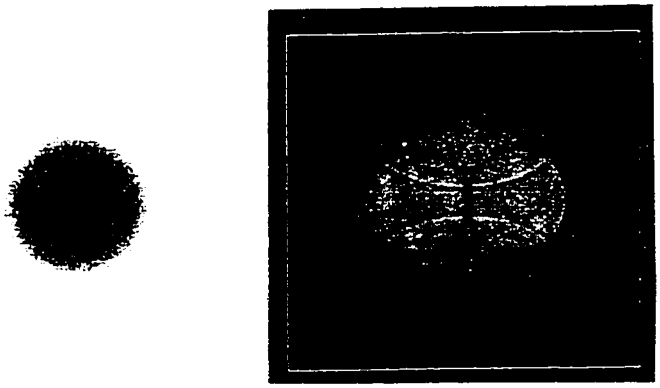


Figure 3.2. A DTF having a form of Gaussian high-pass filter (left) and the corresponding motion corrupted phantom image (right)

form would generate ghosts that are smeared replicas of the central image. However the DTF having high-pass filter form makes ghosts that have the outline of the central image but do not have much detail.

Let's slightly change the profiles of the low-pass filter and the high-pass filter to examine the sensitivity of the ghost appearance to filter characteristics. A DTF was chosen to have the form of Butterworth low-pass filter as:

$$F_{non-rigid}(k_x, k_y) = \alpha/(1 + [\frac{D(k_x, k_y)}{D_0}]^{2m}) \tag{3.9}$$

with D_0 being the cut-off frequency where the transfer function is down 50% from its maximum value. The order of the filter is m. The bigger m is. the sharper the change between the passed and suppressed frequencies becomes. With image size = 256 × 256. $m = 3$. $D_0 = 25.0$ and $\alpha = 0.6$. the DTF having the form of the Butterworth low-pass filter of Equation 3.9 is shown in Fig. 3.3(left). The corresponding motion corrupted phantom image is shown in Fig. 3.3(right).

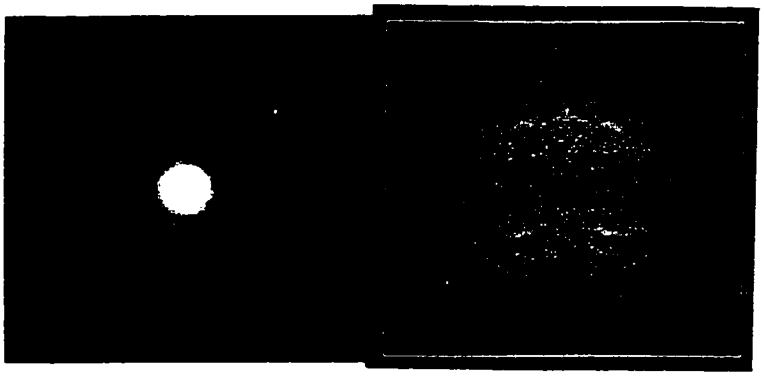


Figure 3.3. A DTF having a form of a Butterworth low-pass filter (left) and the corresponding motion corrupted phantom image (right)

The Butterworth high-pass filter is given by:

$$F_{non-rigid}(k_x, k_y) = \alpha/(1 + [\frac{D_0}{D(k_x, k_y)}]^{2m}) \tag{3.10}$$

The frequency domain image with $m = 3$, $D_0 = 25.0$ and $\alpha = 0.6$ is shown in Fig. 3.4(left). The corresponding motion corrupted phantom image is shown in Fig. 3.4(right).

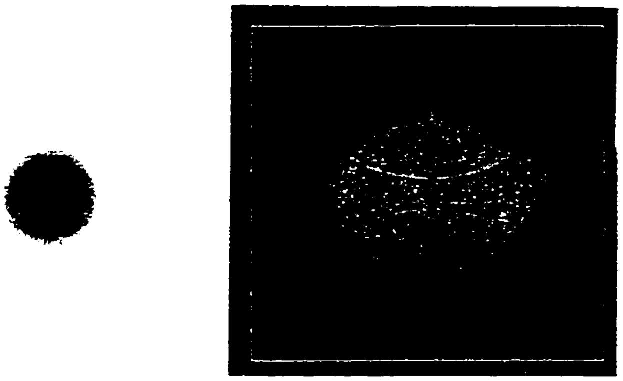


Figure 3.4. A DTF having a form of Butterworth high-pass filter (left) and the corresponding motion corrupted phantom image (right)

Comparing the two phantom images corrupted by two different DTFs having low-pass filter forms in Fig. 3.1 and Fig. 3.3 respectively, we can not tell much difference in the appearance of the ghosts by our eyes. The same is true for the two motion corrupted phantom images in Fig. 3.2 and Fig. 3.4 that use high-pass filter.

Let's see what kind of artifacts we have if the DTF is chosen to have a form of radially symmetric Butterworth band-reject filter. The DTF is given by:

$$F_{non-rigid}(k_x, k_y) = \alpha/(1 + [\frac{D(k_x, k_y)w}{D^2(k_x, k_y) - D_0^2}]^{2m}) \tag{3.11}$$

where w is the width of the band and D_0 is its centre. Fig. 3.5(left) shows an example of this type of DTF when $m = 3$, $D_0 = 45.0$, $w = 35.0$ and $\alpha = 0.6$. The corresponding motion corrupted phantom image is shown in Fig. 3.5(right). The ghosts in Fig. 3.5 have the outlines of the central image and at the same time also keep some detail of the central image.

Let's see the DTF having a form of radially symmetric Butterworth band-pass filter. Mathematically a band-pass filter can be obtained by a "flipping" band-reject

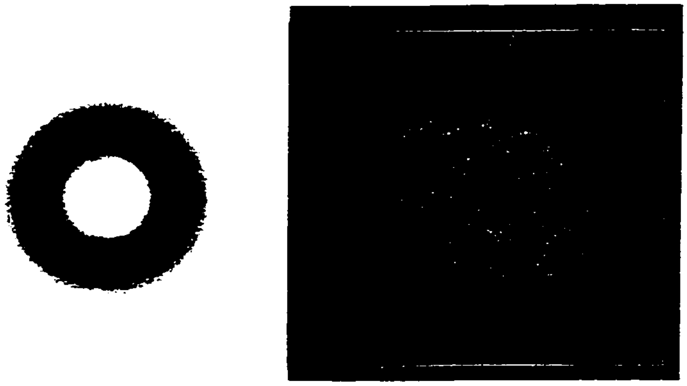


Figure 3.5. A DTF having a form of radially symmetric Butterworth band-reject filter (left) and the corresponding motion corrupted phantom image (right)

filter as:

$$F_{non-rigid}(k_x, k_y) = \alpha \cdot (1 - 1/(1 + (\frac{D(k_x, k_y)w}{D^2(k_x, k_y) - D_0^2})^{2m})) \qquad (3.12)$$

This filter's response with parameters $\alpha = 0.6$, $m = 3$, $D_0 = 45.0$, and $w = 35.0$ is shown in Fig. 3.6(left). Likewise, the corresponding corrupted phantom image is shown in Fig. 3.6(right). The ghosts in Fig. 3.6 have very similar appearance as that in Fig. 3.4 with high-pass filter.

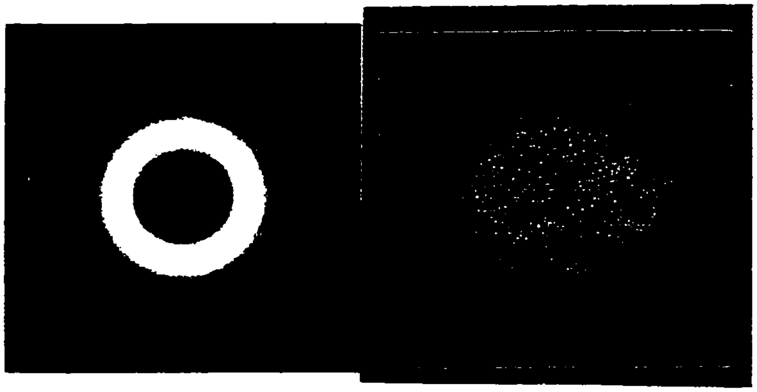


Figure 3.6. A DTF having a form of a radially symmetric Butterworth band-pass filter (left) and the corresponding motion corrupted phantom image (right)

There are alternative ways to manipulate 2D filters. The following are two examples of different versions of Butterworth band-pass filters and Butterworth band-reject filters. Instead of being radially symmetric, the followings are symmetric just to the k_y axis as:

$$F_{non-rigid}(k_x, k_y) = \alpha/(1 + [\frac{k_y w}{k_y^2 - D_0^2}]^{2m}) \qquad (3.13)$$

$$F_{non-rigid}(k_x, k_y) = \alpha \cdot (1 - 1/(1 + [\frac{k_y w}{k_y^2 - D_0^2}]^{2m})) \qquad (3.14)$$

Equation 3.13 is the belt Butterworth band-reject filter along the k_y direction. The frequency data of the DTF and the corresponding corrupted phantom image under the same condition as with the radially symmetric band-reject filter in Fig. 3.5 are shown in Fig. 3.7. Equation 3.14 is the belt Butterworth band-pass filter along the k_y direction. The frequency data of the DTF and the corresponding phantom corrupted image under the same condition as with the radially symmetric band-pass filter in Fig. 3.6 are shown in Fig. 3.8.

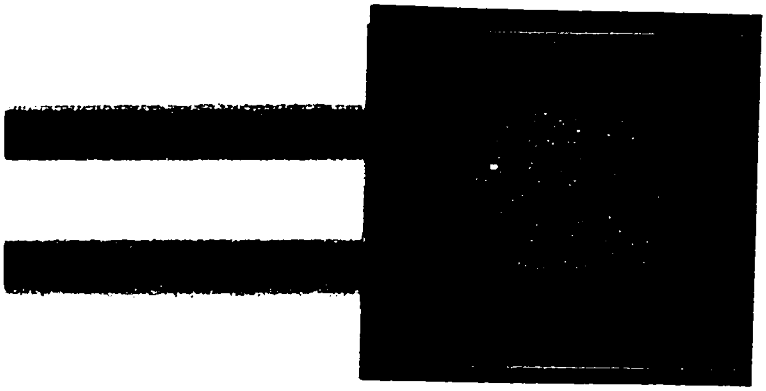


Figure 3.7. A DTF having a form of a belt Butterworth band-reject filter (left) and the corresponding motion corrupted phantom image (right)

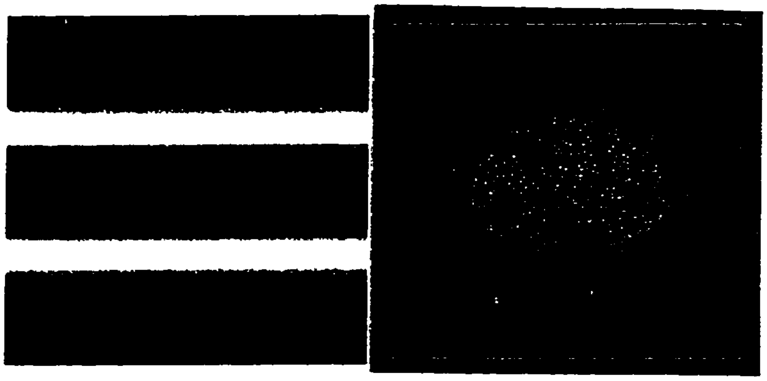


Figure 3.8. A DTF having a form of belt Butterworth band-pass filter (left) and the corresponding motion corrupted phantom image (right)

Compared with ghosts simulated by radially symmetric band-reject and band-pass DTFs, the ghosts generated by belt band-reject and band-pass DTFs have outlines and some detail of the central image in one direction with outlines and some detail in the other direction missing at some places. This characteristics is very close to that in the abdomen image in Fig. 2.5.

Based on the phantom images with simulated motion artifacts shown in Fig. 3.1-Fig. 3.8. the following observation can be drawn:

- The artifacts from different DTFs have a common feature: the ghosts are shifted. scaled and distorted replicas of the central image.
- There is difference in the appearance of ghosts among different phantom images corrupted by different DTFs.
- The artifacts arisen from DTFs having the form of low-pass filters might have been characterized using Mitsa's AM model if only a small amount of smearing is introduced.
- The ghost outlines of the phantom image generated by a DTF having the form of a belt band-pass filter in Fig. 3.8 are closer to that displayed in the abdomen image in Fig. 2.5.

3.5 Summary

In this chapter first we point out that assuming every point in the imaged subject moves identically is not appropriate for real MRI cases. Then a more realistic generalized motion model is proposed. The spatially variant characteristics of motions of the imaged subject is taken into account by introducing a distortion transfer function (DTF) into the generalized motion model. The existing motion models are shown to be just special cases of this generalized motion model. Finally phantom images simulated by using DTFs of the form of different types of filters are shown to have different ghost appearances. The ghost artifacts arisen from DTFs having the form of low-pass filters might have been characterized using Mitsa's AM model if less smearing introduced. The ghost outlines of the phantom image generated by a DTF having the form of a belt band-pass filter as shown in Fig. 3.8 are closer to those displayed in the abdomen image in Fig. 2.5. In the next chapter. we shall discuss the theory and implementation of contour detection by the snakes and region labeling using the snake contours. which are steps needed before we can determine the DTFs for images with ghosts and remove the ghost artifacts.

CHAPTER 4

THE THEORY OF SNAKES

4.1 Introduction

Chapter 4. 5. 6 contain theories and implementations about how to detect contours and how to segment an image into meaningful regions. The main reasons why contour detection and image segmentation are needed in this thesis are:

1. In Chapter 7, the estimation of distortion transfer functions (DTFs) requires information about contours of the central image and ghosts.

2. As will be shown in Chapter 8, our composite image processing method to suppress motion artifacts requires segmentation information about the regions of support (ROSs) of the central image and ghosts and how these regions of support overlap with each other.

The discussion in this chapter begins with basic image segmentation and contour detection techniques in general medical imaging applications. It is then shown that the conventional mask operators are not suitable for the motion corrupted abdomen image. So that a new theory of contour detection technique. the snake. is presented. The description of snake theory in this Chapter is mainly based on published papers [Lobregt and Viergever 1995; Kass et al. 1988; Cohen 1991; Ranganath 1995; Wang et al. 1996; Amini et al. 1990; Staib and Duncan 1992]. The application of snakes for segmentation will be discussed in Chapter 5 and 6.

4.2 Contour Detection Techniques in General Medical Imaging Applications

Generally in computer vision there are two approaches for segmentation: The first approach is to locate boundaries or edges of regions. The second is to group points into similar regions. which then permits determination of boundaries. The second approach is not feasible for the MRI motion corrupted images as the central image and ghosts overlap with each other and the points inside regions do not have unique objective properties. By using the first approach, the segmentation problem turns into two steps:

- First. how to get boundaries or, equivalently. how to detect contours;
- Second, how to get regions from boundaries. or how to label regions inside contours.

In general image processing, contour detection is the first step of many image processing tasks. such as automatic analysis, image matching. motion tracking. computer vision and three-dimensional reconstruction. In medical imaging applications. there are different techniques to accomplish object contour detection. The techniques can be put into four categories according to Lobregt and Viergever [1995]:

- Completely manual way. A technician will sit before a screen and use a pointing device (a mouse or a graphics tablet) to draw the desired contour point by point. To do the job, the technician has to have expert knowledge about the clinical problem, as well as a certain level of skill in using the available pointing device. As we can imagine, manual definition of contour is a time consuming and boring process, which presents a serious bottleneck in processing large data sets. Moreover, manual contour definition suffers from a very poor reproducibility.
- Fully automatic way. An algorithm run in computer is used to be able to extract the desired contour automatically. Unlike the completely manual way. the contours by fully automatic way are reproducible. However so far the available techniques of contour detection work only when contour information of images is not very complex. For example when the object's contour is very clear and

strong and there is little interference from undesired structures and noise. For many real medical applications, the existing techniques are not good enough for fully automatic contouring.

- The way of automated first guess, followed by manually editing. Usually a rough guess of the contour is obtained by means of some simple techniques such as thresholding and region growing. Then a technician modifies the obtained contour manually using not only the information displaying in the image but also the contextual knowledge about the object's anatomy and pathology. As with the completely manual way, the disadvantages of being time consuming and boring are still there.

- The way of manual rough guess, followed by automatic contour detection. At the beginning, a rough contour is defined manually using some kind of pointing device. Then the initial contour is refined automatically by an algorithm. Normally the influence of the operator on final result is indirect. The disadvantages of manual approaches are reduced while the advantage of full automatic way is kept when the same final contour results from various initial contours.

In this project, a potential fully automatic way, the conventional mask operators, was first tried. After the mask operators were found to fail to work for the real abdomen image, a new technique, the snake was employed. The snake contour detection belongs to category four: manual rough guess followed by an automatic algorithm.

4.3 Mask Operators

Among all the conventional mask operators, Sobel mask operators are very popular and considered very sensitive to edges [Levine 1985]. The 3 × 3 Sobel mask operators used for detecting edges normal to the eight compass directions are shown in Fig. 4.1. Fig. 4.2(right) gives the contours detected by a combination of these eight mask operators for the abdomen image (left). From Fig. 4.2(right) we can see clearly that:

- In the abdomen image, the strengths of edges vary largely. At some places, such as the left and right side contours, the edges are very strong. At some other places, such as the bottom contour, the edges are very weak and even broken.
- The structures of the inner parts of the abdomen interfere with the outer boundary of the abdomen. Therefore it is difficult to get the outer boundary of the abdomen by thresholding strengths of edges.

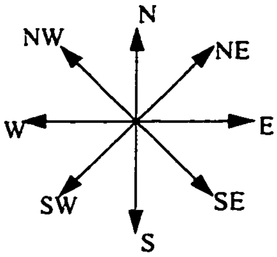
In order to get the continuous boundaries for real motion corrupted MR images. a new contour detection method: the snake, is suggested.

4.4 Continuous Active Contour Model

According to [Lobregt and Viergever 1995: Kass et al. 1988: Cohen 1991: Ranganath 1995; Wang et al. 1996; Amini et al. 1990; Staib and Duncan 1992]. the snake. also called an active contour model, is an energy-minimizing spline influenced by external constraint forces and internal smoothness constraint forces and guided by image forces that pull the contour toward image features such as edges. In other words, besides controlled by the edges (image forces), the snake contour can be manipulated and constrained by some other forces, such as *a priori* knowledge about the contour (external forces) and smoothness and sharpness constraints about the contour (internal forces). The snake paradigm models a deformable contour as possessing image energy, external energy and internal energy. When the contour is acted on by these three energy fields, the contour seeks equilibrium at a minimum of total energy field by moving and changing shape. That is how the technique gets the name "snake".

In the remainder of this chapter, we adopt the notations used in Kass *et al.*'s [1988] and Ranganath's [1995] to mathematically describe the continuous active contour

The compass directions:



Sobel masks for edges normal to the compass directions
(N, NW, W, SW, S, SE, E, NE)

$$\text{Sobel}_1 \ \text{(N)} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad \text{Sobel}_2 \ \text{(NW)} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

$$\text{Sobel}_3 \ \text{(W)} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad \text{Sobel}_4 \ \text{(SW)} \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & i & 0 \end{bmatrix}$$

$$\text{Sobel}_5 \ \text{(S)} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad \text{Sobel}_6 \ \text{(SE)} \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

$$\text{Sobel}_7 \ \text{(E)} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \text{Sobel}_8 \ \text{(NE)} \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

Figure 4.1. The Sobel mask operators

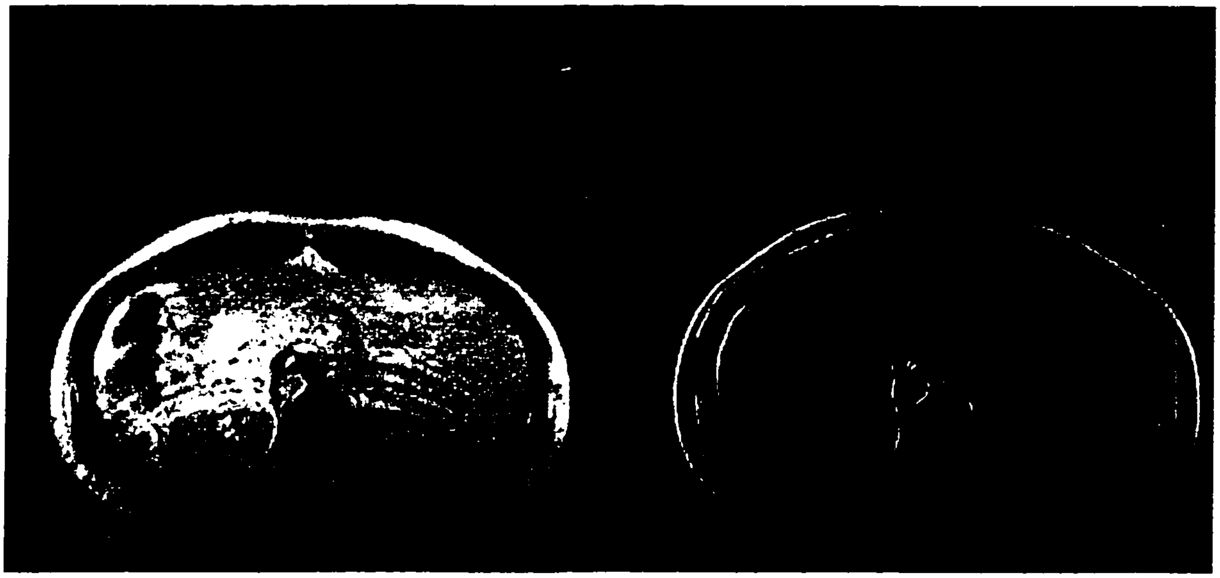


Figure 4.2. The contours (right) of the abdomen image (left) detected by the Sobel mask operators. Note the varying strength of the outer boundary.

model. Assume a contour is represented parametrically as: $\mathbf{V}(s) = (Vx(s). Vy(s))$, where Vx and Vy are the x and y coordinates respectively of any point on the contour and s is the parameter. Conventionally the value of s is chosen to change from 0 to 1 when the contour goes from the beginning to the end. The energy of the contour is defined as:

$$\begin{aligned} E_{snake-total} &= \int_0^1 E_{snake}(\mathbf{V}(s))ds \\ &= \int_0^1 (\ E_{int}(\mathbf{V}(s)) + E_{ext}(\mathbf{V}(s)) + E_{img}(\mathbf{V}(s))\ )ds \qquad (4.1) \end{aligned}$$

where E_{int} represents the internal force (or smoothness constraints) which may have both stretching and bending terms, E_{img} stands for the image force constructed from some image features to attract the snake into, E_{ext} is the force due to extraneous constraints which can be derived from knowledge bases and/or users.

The strategy of snakes is to deform the contour to the locations that make the above total energy a minimum. A contour that fits the image features. *a priori* knowledge and smoothness constraints can be obtained by balancing the internal forces. the external forces and the image forces. We summarize the advantages of snakes as:

- The energy field provides a flexible mechanism to incorporate information from different sources and combine them to extract the boundaries.

- Contour connectivity is part of the snake model due to the smoothness constraints and the integration of energy along entire length of the contour as discussed in more detail later. Therefore the snake model has the potential advantage of robustness to image and noise. By using snakes, the problem suffered by many other contour extraction techniques is overcome. For example the Sobel mask operators, which use only local edge information, often result in broken edges when edges are not strong enough at some places.

These attractive properties make snakes useful in many medical applications, such as contour extraction for cardiac MRI studies [Ranganath 1995].

The three energy fields associated with the snakes algorithm are discussed one by one in the following sections.

4.4.1 Internal Energy

The following internal energy expression is widely used in literature[Lobregt and Viergever 1995; Kass et al. 1988; Cohen 1991; Ranganath 1995; Wang et al. 1996; Amini et al. 1990; Staib and Duncan 1992]:

$$\begin{aligned} E_{int} &= \frac{1}{2}(\tilde{\alpha}(s)\mathbf{V}_s^2(s) + \tilde{\beta}(s)\mathbf{V}_{ss}^2(s)) \\ &= \frac{1}{2}\tilde{\alpha}(s)(\mathbf{V}^2x_s(s) + \mathbf{V}^2y_s(s)) + \frac{1}{2}\tilde{\beta}(s)(\mathbf{V}^2x_{ss}(s) + \mathbf{V}^2y_{ss}(s)) \qquad (4.2) \end{aligned}$$

where $\mathbf{V}_s(s) = \partial\mathbf{V}/\partial s$, $\mathbf{V}_{ss}(s) = \partial^2\mathbf{V}/\partial s^2$, $\mathbf{V}x_s(s) = \partial\mathbf{V}x/\partial s$, $\mathbf{V}y_s(s) = \partial\mathbf{V}y/\partial s$, $\mathbf{V}x_{ss}(s) = \partial^2\mathbf{V}x/\partial s^2$, $\mathbf{V}y_{ss}(s) = \partial^2\mathbf{V}y/\partial s^2$. $\tilde{\alpha}(s)$, $\tilde{\beta}(s)$ are weight coefficients.

As described by Kass *et al.* [1988], the energy is composed of a first-order term controlled by $\tilde{\alpha}(s)$ and a second-order term controlled by $\tilde{\beta}(s)$. These two items make the snake act like a membrane or like a thin plate respectively [Kass et al. 1988]. Adjusting the weights $\tilde{\alpha}(s)$ and $\tilde{\beta}(s)$ controls the relative importance of the membrane and thin-plate terms. A large $\tilde{\alpha}(s)$ penalizes the development of positional discontinuities and encourages the contour's tendency to shrink [Ranganath 1995]. Hence $\tilde{\alpha}(s)$ is also called the elasticity coefficient. Similarly a large $\tilde{\beta}(s)$ discourages sharp bends in the contour [Ranganath 1995]. Setting $\tilde{\beta}(s)$ to zero allows the snake to become second-term discontinuous and develop a corner [Kass et al. 1988]. Thus $\tilde{\beta}(s)$ is also called the rigidity coefficient.

4.4.2 Image Energy

In order to make snakes useful we need energy functionals that attract them to desired features in images. There are different image energy functionals which can be used. The total image energy can be expressed as a weighted combination of different energy functionals. By adjusting the weights, a wide range of snake behavior can be created. Kass *et al.* [1988] gave two examples of image energy functionals as:

- Line Functional. The simplest image functional is the image intensity $I(\mathbf{V}(s))$ itself. If we set:

  $$E_{line}(\mathbf{V}(s)) = I(\mathbf{V}(s)) \cdot w_{line}(\mathbf{V}(s))$$

  then depending on the sign of weight w_{line}, the snake will be attracted either to light lines or dark lines. Subject to its other constraints, the snake will try to align itself with the lightest or darkest nearby lines. This energy functional can be used when the desired contour is a line shape.

- Edge Functional. Finding edges in an image can be done with a simple energy functional as:

  $$E_{edge}(\mathbf{V}(s)) = -| \bigtriangledown I(\mathbf{V}(s))|^2$$

  This causes the snake to be attracted to contours with large image gradients.

  When only the features of lines and edges are considered, the image energy is the combination of the line functional and edge functional as:

  $$E_{img}(\mathbf{V}(s)) = E_{line}(\mathbf{V}(s)) + E_{edge}(\mathbf{V}(s))$$

4.4.3 External Energy

Besides the internal energy and image energy, the external energy takes the extraneous constraints into account. *a priori* knowledge is a common source of extraneous constraints. For example we can manually give the snake some point locations which the final contour has to pass through.

4.5 Calculation of the Total Energy for the Discretized Model

Based on the above discussion about internal energy, image energy and external energy, the total energy is given by:

$$\begin{aligned} E_{snake-total} &= \frac{1}{2}\int_0^1 \tilde{\alpha}(s)(\mathbf{V}^2 x_s(s) + \mathbf{V}^2 y_s(s))ds + \frac{1}{2}\int_0^1 \beta(s)(\mathbf{V}^2 x_{ss}(s) + \mathbf{V}^2 y_{ss}(s))ds \\ &+ \int_0^1 (E_{ext}(\mathbf{V}(s)) + E_{img}(\mathbf{V}(s)))ds \qquad (4.3) \end{aligned}$$

The desired contour can be found by minimizing the above total energy.

The above energy equation is for continuous cases. For digital images, Equation 4.3 has to be discretized. The standard $O(h^2)$ derivative approximations are widely used in the literature [Lobregt and Viergever 1995; Kass et al. 1988; Cohen 1991; Ranganath 1995; Wang et al. 1996; Amini et al. 1990; Staib and Duncan 1992]. The notations used in this section follow Ranganath [1995].

$$E_{snake-total} = \frac{1}{2h}\sum_{i=0}^{N-1} \tilde{\alpha}_i(|\mathbf{V}_i - \mathbf{V}_{i-1}|)^2 + \frac{1}{2h^3}\sum_{i=0}^{N-1} \tilde{\beta}_i(\mathbf{V}_{i-1} - 2\mathbf{V}_i + \mathbf{V}_{i+1})^2 + h\sum_{i=0}^{N-1} E_{ie}(\mathbf{V}_i) \qquad (4.4)$$

where h is the space step of the discretization and the contour is uniformly sampled at N points along the contour, $\mathbf{V}_i = \mathbf{V}(s)|_{s=ih}$, $E_{ie} = E_{img} + E_{ext}$.

To get the desired contour that minimizes the above energy, we set $\partial E_{snake-total}/\partial \mathbf{V}_i$, $i = 0, 1, \ldots, N-1$, to zero. Thus:

$$\begin{aligned} -\frac{\partial E_{ie}}{\partial \mathbf{V}_i} &= \mathbf{V}_i(\alpha_i + \alpha_{i+1} + \beta_{i-1} + 4\beta_i + \beta_{i+1}) \\ &+\mathbf{V}_{i-1}(-\alpha_i - 2\beta_{i-1} - 2\beta_i) \\ &+\mathbf{V}_{i+1}(-\alpha_{i+1} - 2\beta_i - 2\beta_{i+1}) \\ &+\mathbf{V}_{i-2}(\beta_{i-1} + v_{i+2}\beta_{i+1}); \quad 0 \leq i \leq N-1 \end{aligned} \tag{4.5}$$

where $\alpha_i \equiv \tilde{\alpha}_i/h^2$, $\beta \equiv \tilde{\beta}_i/h^4$.

If we consider closed contours with periodic boundary conditions: $V_0 = V_N$, $V_{-1} = V_{N-1}$, the above equation can be compactly written in matrix notation as

$$\mathbf{AV} + \mathbf{F} = \mathbf{0} \tag{4.6}$$

where

$$\frac{\partial E_{ie}}{\partial \mathbf{V}} = [\frac{\partial E_{ie}}{\partial \mathbf{V}x} \quad \frac{\partial E_{ie}}{\partial \mathbf{V}y}] = [f_{Vx} \quad f_{Vy}] = \mathbf{F} \tag{4.7}$$

The coefficient matrix $\mathbf{A}$ is a pentadiagonal matrix and composed of entries from the coefficients of $\mathbf{V}_{i-2}$ to $\mathbf{V}_{i+2}$ in Equation 4.5. Equation 4.6 is simply a statement of the fact that, at equilibrium, the combined external force and image force $\mathbf{F}$ balances the internal snake forces $\mathbf{AV}$.

4.6 The Evolution Solution of Snakes

The final contour has to meet Equation 4.6. As it is hard to directly solve Equation 4.6, an alternative approach is used. We can make Equation 4.6 dynamic and more general by assigning mass density to the snake and allowing the kinetic energy developed to be dissipated by friction [Ranganath 1995]. According to Ranganath [1995], the dynamic equation corresponding to Equation 4.6 can be written as:

$$\tilde{\mu}\mathbf{V}_{tt} + \tilde{\gamma}\mathbf{V}_t + \mathbf{AV} + \mathbf{F} = 0$$

where t represents time of evolution, $\tilde{\mu}$ is the mass density and $\tilde{\gamma}$ is the dissipative constant. $\mathbf{V}_{tt} = \partial^2\mathbf{V}/\partial t^2$, $\mathbf{V}_t = \partial\mathbf{V}/\partial t$.

Discretizing the above equation in time by using $O(h^2)$ derivative approximations evaluated at $(n-1)\Delta t$ and rearranging, we get:

$$-\mathbf{F}(\mathbf{V}_{n-1}) = (\mathbf{A} + \mu\mathbf{I} + \gamma\mathbf{I})\mathbf{V}_n - (2\mu + \gamma)\mathbf{V}_{n-1} + \mu\mathbf{V}_{n-2}$$

where we have defined $\mu = \tilde{\mu}/(\Delta t)^2$ and $\gamma = \tilde{\gamma}/2\Delta t$. Normally μ is set to zero for simplicity. Therefore the above equation can be written as:

$$(\mathbf{I} + \tau\mathbf{A})\mathbf{V}_n = \mathbf{V}_{n-1} + \tau\mathbf{F}(\mathbf{V}_{n-1}) \quad \mathit{or} \quad \tilde{\mathbf{A}}\mathbf{V_n} = \mathbf{V}_{n-1} + \tau\mathbf{F}(\mathbf{V}_{n-1}) \qquad (4.8)$$

where $\tau = 1/\mu$ is called the step size in time of evolution. $\mathbf{V}_n$, $\mathbf{V}_{n-1}$ and $\mathbf{V}_{n-2}$ are the sampled point coordinates at time $n\Delta t$, $(n-1)\Delta t$ and $(n-2)\Delta t$ respectively. In addition we have:

$$\tilde{\mathbf{A}} = \mathbf{I} + \tau\mathbf{A} \qquad (4.9)$$

when Equation 4.8 stabilizes and the term $\partial\mathbf{V}/\partial t$ tends to zero with $\mathbf{V}_n = \mathbf{V}_{n-1}$, the solution of Equation 4.8 is also the solution to the static problem in Equation 4.6.

4.7 LU Decomposition and Linear Equation System Solving

Equation 4.8 is actually a linear equation system. To solve a linear equation system, LU decomposition method is widely used.

In this section the LU decompositon method is introduced in order to solve the linear equation system described in Equation 4.8. The notations in this section are based on [Cheney and Kincaid 1994].

Assume a system of n linear equations in n unknowns is represented in compact form as:

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad 1 \le i \le n$$

The same system can be written in a matrix form as:

$$\tilde{\mathbf{A}}\mathbf{x} = \mathbf{b}$$

where coefficient elements a_{ij} form a $n \times n$ square matrix $\tilde{\mathbf{A}}$. the unknowns x_i and the right-hand side elements b_i form two $n \times 1$ vectors $\mathbf{x}$ and $\mathbf{b}$ respectively.

The theory of linear equation system shows that the $n \times n$ matrix $\tilde{\mathbf{A}}$ can be decomposed into a product of two simple matrices: one unit lower triangular matrix $\mathbf{L}$ and the other upper triangular matrix $\mathbf{U}$ as:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & . \; . \; . & 0 \\ l_{21} & 1 & 0 & . \; . \; . & 0 \\ l_{31} & l_{32} & 1 & . \; . \; . & 0 \\ . & . & . & . & . \\ . & . & . & \;\; . & . \\ . & . & . & \;\;\;\; . & . \\ l_{n1} & l_{n2} & l_{n3} & . \; . \; . & 1 \end{bmatrix}$$

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & . \; . \; . & u_{1n} \\ 0 & u_{22} & u_{23} & . \; . \; . & u_{2n} \\ 0 & 0 & u_{33} & . \; . \; . & u_{3n} \\ . & . & . & . & . \\ . & . & . & \;\; . & . \\ . & . & . & \;\;\;\; . & . \\ 0 & 0 & 0 & . \; . \; . & u_{nn} \end{bmatrix}$$

In short. an LU decomposition of matrix $\tilde{\mathbf{A}}$ is refered to as:

$$\tilde{\mathbf{A}} = \mathbf{LU}$$

The unit lower triangular matrix $\mathbf{L}$ and the upper triangular matrix $\mathbf{U}$ can easily be obtained from the forward naive Gaussian elimination process. The matrix $\mathbf{L}$ consists of all the multipliers located in the positions of the elements they annihilated from $\tilde{\mathbf{A}}$, of unit diagonal elements and of 0 upper triangular elements. The matrix $\mathbf{U}$ is upper triangular (not necessarily unit diagonal) and is the final coefficient matrix after the forward elimination phase is completed.

Using LU decomposition, the linear system $\tilde{\mathbf{A}}\mathbf{x} = \mathbf{b}$ can be rewritten in another way as:

$$\mathbf{LUx} = \mathbf{b}$$

Then the problem is simplified to solve two simple triangular systems:

$$\mathbf{Lz} = \mathbf{b}, \quad \mathbf{Ux} = \mathbf{z}$$

z can be obtained by pseudo code as:

```
$z_1 = b_1$
for( i from 2 to n)
        $z_i = b_i - \sum_{j=1}^{i-1} l_{ij} z_j$
endfor
```

Likewise. **x** can be obtained by pseudo code as:

```
$x_n = \frac{z_n}{u_{nn}}$
for ( i from n-1 to 1 by step -1)
        $x_i = (z_i - \sum_{j=i+1}^{n} u_{ij} x_j)/u_{ii}$
endfor
```

4.8 Summary

The strategy of snakes is to locate positions of a minimum energy field. The energy field is actually an integration of several different kinds of energy along the entire contour. There are usually three kinds of energy: image energy from image features, external energy from extraneous knowledge or constraints and internal energy from the smoothness constraints about the contour. The advantages of snakes include: flexible mechanism to incorporate all kinds of information about the boundary; contour connectivity; and robustness to noise. For computation. discretized snake models are required. The solution of a discretized snake model can be obtained by solving a linear equation system evolutionarily. A method, called LU decomposition method. for solving linear equation systems is also described.

In the next chapter we will use these concepts involved in this chapter to implement the snake algorithm.

CHAPTER 5

THE IMPLEMENTATION OF THE SNAKE ALGORITHMS

5.1 Introduction

The previous chapter (The Theory of Snake) was a summary of published work [Lobregt and Viergever 1995; Kass et al. 1988; Cohen 1991: Ranganath 1995: Wang et al. 1996; Amini et al. 1990; Staib and Duncan 1992]. Although presentation on the theory behind the snake is done well in these papers, considerable work was necessary to modify the algorithms to work for motion corrupted MRI images. The block diagram and procedures of our implementation are first given in this chapter. Then considerations about the implementation are discussed in detail, including:

- contour representation;
- contour resampling;
- determination of snake coefficients;
- stabilities of snakes.

Then it is shown how to apply our implementation of the snake to detect contours of the central image for the motion corrupted abdomen image in Fig. 2.5. Finally a ballooning algorithm is suggested to be used for obtaining contours of ghosts.

5.2 The Block Diagram of Our Implementation of Snakes

Since details of other authors' implementation of the snake algorithms are not provided in their papers, we have given detailed information of our algorithms.

The block diagram of our implementation of snakes is shown as Fig. 5.1. The steps of this implementation follow. Additional detail is given in later sections of this chapter.

1. Input all the required parameters and coefficients: the elasticity coefficient α_i and the rigidity coefficient β_i in Equation 4.5, the step size τ in Equation 4.8, the desired length l_{des} between two consecutive evenly spaced points on a contour for the resampling algorithm, and the cut-off frequency of the low-pass filter for smoothing the input image.

2. Use the low-pass filter to smooth the image and thus the edges of the image. This smearing provides more flexibility to choose the initial contour. That is the initial contour does not have to be exactly placed close to the desired contour.

3. Input the number of starting points M^i ($i = 0$ at this time) and the starting points P_0^i, P_1^i, ... , $P_{M^i-1}^i$, which are supposed to be in the vicinity of the desired contour. These points are usually obtained manually and do not have to be evenly spaced.

4. Resample the M^i points P_0^i, P_1^i, ... , $P_{M^i-1}^i$ to obtain N^i equally spaced points V_0^i, V_1^i, ... , $V_{N^i-1}^i$ in order to gather information evenly along the contour.

5. Compose the $N^i \times N^i$ matrix $\tilde{A}$ according to Equation 4.5.

6. Calculate the image forces at the N^i locations of the evenly spaced points along the contour and obtain the image force vector F according to Equation 4.7.

7. Use the LU decomposition method to solve the linear equation system $\tilde{\mathbf{A}}\mathbf{P}^{i+1} = \mathbf{V}^i + \tau\mathbf{F}(\mathbf{V}^i)$ as presented in Equation 4.8. The solution is N^i new points P_0^{i+1}, P_1^{i+1}, ... , $P_{N^i-1}^{i+1}$. These points determine a new contour which is supposed to be closer to the desired contour. The new points may be unevenly spaced.

8. stop after a given iteration time or when some other criteria is met. Otherwise set $M^{i+1} = N^i$ and treat these N^i points P_0^{i+1}, P_1^{i+1}, ... , $P_{N^i-1}^{i+1}$ as intial points

and go back to step 4 for the $(i+1)^{th}$ iteration.

All these steps are discussed below in detail in the remainder of this chapter.

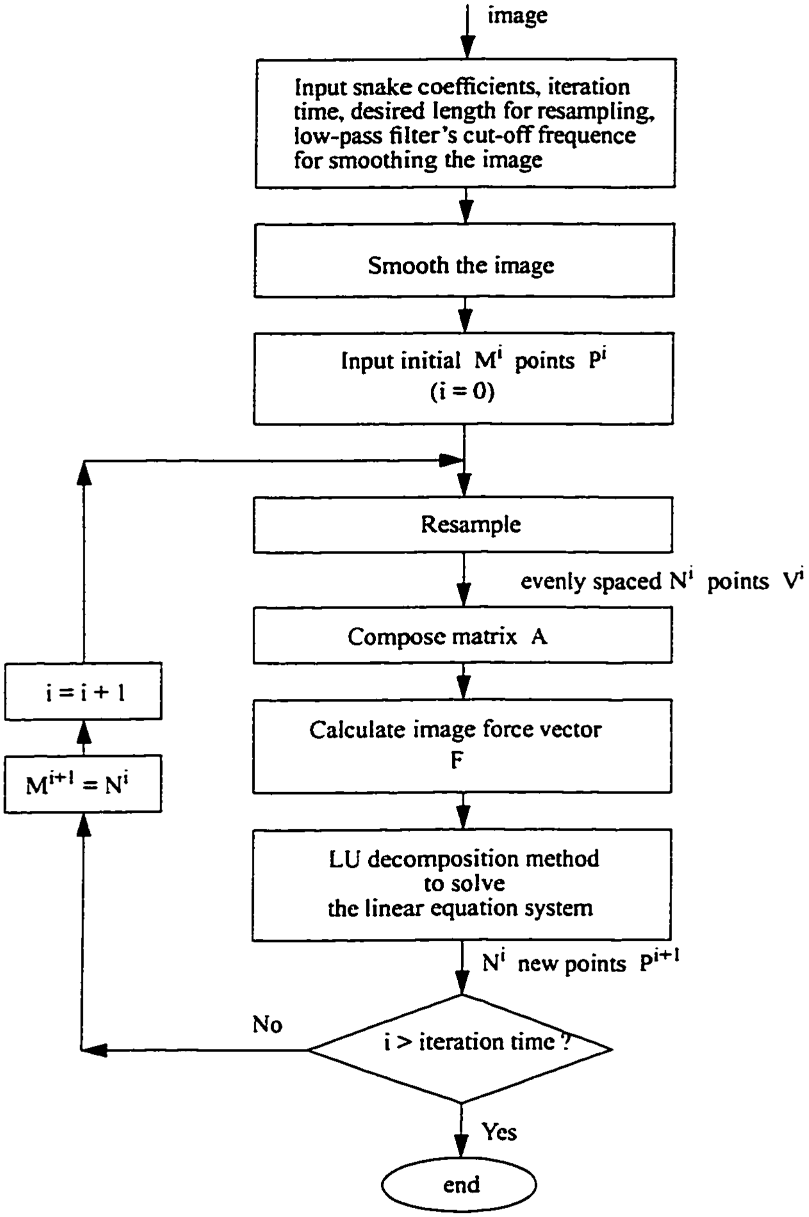


image
Input snake coefficients, iteration time, desired length for resampling, low-pass filter's cut-off frequence for smoothing the image
Smooth the image
Input initial M^i points P^i (i = 0)
Resample
evenly spaced N^i points V^i
Compose matrix A
Calculate image force vector F
LU decomposition method to solve the linear equation system
N^i new points P^{i+1}
i > iteration time ?
No
Yes
end
i = i + 1
M^{i+1} = N^i

Figure 5.1. The block diagram of our implementation of the snake

5.3 Contour Representation

At the beginning of a snake algorithm, we only have a few manually located points along the initial contour. How do we represent the contour given points along it? According to Foley *et al.* [1996], starting at an arbitrary point on the contour and tracing once around it yields a sequence of points: $P_0, P_1, ..., P_{N-1}$. A typical text file for representing a contour has the form as N, (x_0, y_0), (x_1, y_1), (x_2, y_2), ... , (x_{N-1}, y_{N-1}), where N is the number of sampled points on the contour. $(x_i, y_i), i = 0, ..., N-1$, are coordinates in the horizontal and vertical directions respectively. For digital images in computers, the coordinate system is usually an integer Cartesian coordinate system. The pixels lie at intersections of the grid lines.

To display a contour corresponding to the text file N, (x_0, y_0), (x_1, y_1), (x_2, y_2), ... , (x_{N-1}, y_{N-1}), an image array needs to be constructed. For example, if the points on the contour are flagged by a flag value and other points are set to zero, the image array for displaying the contour can be constructed by the following pseudo code as:

```
for(y from top to bottom)
    for (x from left to right)
        Intensity(x,y) = 0;
    endfor
endfor
for (contour point i from 0 to N-1)
        Intensity($x_i, y_i$) = flag value
endfor
```

In this thesis project, starting points along initial contours for snakes are drawn manually by using the image displaying program called "Viewdiff" developed in our lab.

5.4 The Smoothed Image Gradient

As discussed in section 5.2, before the snake evolution begins. a low-pass filter is applied to smooth the image. That is because in this project we found when the desired edges or boundaries were very sharp and gradients a distance away from the sharp edges were almost zero, the snake algorithm was provided with no clue about which direction led to the maximum gradients or minimum energy. Moreover sometimes when the seed contour was a fairly large distance away from the desired contour. the evolution may converge to an undesired local energy minimum position. To allow snakes to be attracted to the equilibrium from far away, image gradients need to be smoothed. This image smoothing can be done in frequency domain by a low-pass filter. The low-pass filter used in this project is a Gaussian filter as:

$$H(u.v) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{u^2+v^2}{2\sigma^2})$$

where σ is the standard deviation. We typically used σ values 10.0-80.0 for images of size 256 by 256 depending on how much smearing was required. A small σ leads to a narrow filter bandwidth and a more blurred image edges.

With the image energy so blurred. the snakes were attracted to the energy minimum direction from very far away. but sometimes did a poor job of localizing the edges. In this case, we can repeatedly run the snake algorithm more than once. That is the output points of this run is sent to the input of the next run with a reduced σ. Slowly reducing the blurring can lead the snake finally to the accurate energy minimum position.

5.5 Contour Resampling

As discussed in section 5.2, step 4 of our snake implementation is resampling. The goal of resampling is to obtain the evenly spaced points along a contour needed for discrete snake algorithms given some randomly spaced points. The issue of resampling is mentioned in the literature about snakes. However only one paper states in some detail how the resampling is done. In [Lobregt and Viergever 1995], the resampling is implemented as a two-pass process. The first pass checks along the entire contour if any segment length has become shorter than a given minimum length. If this is the case, this edge segment is removed from the contour by replacing the two vertices on both ends of this segment by one single vertex at a position exactly in between the replaced vertices as shown in Fig. 5.2(a). The second pass checks again along the entire contour, but now for segments with length larger than a given maximum length. Such an edge segment is divided into two shorter ones of equal length as shown in Fig. 5.2(b).

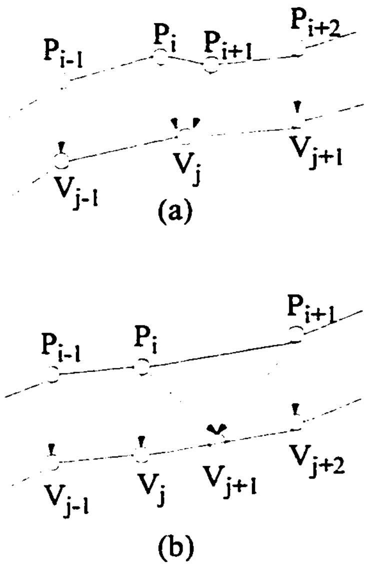


Figure 5.2. Illustration of the existing contour resampling [Lobregt and Viergever 1995]. (a) When the length between two adjacent points is less than a fixed minimum length, replace the two points by one point. (b) When the length between two adjacent points is bigger than a fixed maximum length, insert a new point between the two points

The contour resampling of [Lobregt and Viergever 1995] is easily demonstrated to be very weak. First oscillatory behavior can happen in which the vertices are repeatedly removed in one resampling action and inserted again in the next. Second the length between two resampled points changes from point to point and is not equal to the desired length l_{des}. In other words, the resampled points are not evenly spaced at exactly the same interval l_{des}. To solve these two problems. a new resampling algorithm was developed for this thesis project.

Let's discuss our new resampling algorithm. Assume we have M starting points along a contour. The M starting points are ordered as: $P_0, P_1, ..., P_{M-1}$. where P_0 is the first point at the beginning, P_{M-1} is the last point at the end. Since the contour is closed. we have $P_M = P_0$, $P_{-1} = P_{M-1}$. It does not matter which point is treated as the first point. Px_i and Py_i are the x and y coordinates corresponding to the i^{th} starting point $P_i(i = 0, 1, ..., M-1)$. The idea of our resampling algorithm is not just to adjust the starting points by removing and inserting as in [Lobregt and Viergever 1995]. but to resample all the points along the segments P_0P_1, P_1P_2, ..., $P_{M-2}P_{M-1}$ and $P_{M-1}P_0$ and make the length between two adjacent resampled points equal to l_{des}. We start from point P_0. The first resampled point V_{00} along the first segment between P_0P_1 is simply chosen to be P_0. There are n_0 more resampled points along this first segment. where n_0 is determined by:

$$n_0 = trunc(l_0/l_{des})$$

where $trunc()$ is the function of rounding a float number towards the nearest integer that is not bigger than this float number. l_0 is the length of this first segment and equal to the distance between P_0 and P_1.

The next n_0 resampled points $V_{01}, V_{02}, ... , V_{0n_0}$ lie on the segment P_0P_1. Their coordinates can be determined by the following equation as:

$$\frac{Vx_{0k} - Px_0}{Px_1 - Px_0} = \frac{Vy_{0k} - Py_0}{Py_1 - Py_0} = \frac{k * l_{des}}{l_0}, \qquad k = 1, ..., n_0$$

As l_0 is often not equal to integer times of l_{des}, there is a fragment left between the resampled point V_{0n_0} and P_1. This fragment, denoted as l_{E0}, is equal to $rem(l_0/l_{des})$

and less than l_{des}, where $rem(a/b)$ is the function of obtaining remainder of division a/b. The fragment l_{E0} is suppose to be absorbed when we resample the next segment P_1P_2. At the beginning of segment P_1P_2, we assign another fragment denoted by l_{B1} to make $l_{B1} + l_{E0} = l_{des}$, where l_{B1} is also less than l_{des}. Thus the coordinate of the first resampled point V_{10} along segment P_1P_2 is determined by l_{B1} as:

$$\frac{Vx_{10} - Px_1}{Px_2 - Px_1} = \frac{Vy_{10} - Py_1}{Py_2 - Py_1} = \frac{l_{B1}}{l_1}$$

where $l_{B1} = l_{des} - l_{E0}$. The length l_1 of segment P_1P_2 is equal to $\sqrt{(Px_1 - Px_2)^2 + (Py_1 - Py_2)^2}$.

Assume the length of the segment P_iP_{i+1} is l_i, $i = 0, 1, ..., M-1$. The fragment left by the previous segment $P_{i-1}P_i$ is $l_{E(i-1)}$. If l_i is not big enough to satisfied $l_i \geq l_{des} - l_{E(i-1)}$, there are no resampled points along the segment. The fragment l_{Ei} needed to be absorbed by the next segment is increased to $l_{E(i-1)} + l_i$; Otherwise if l_i is big enough to satisfy $l_i \geq l_{des} - l_{E(i-1)}$, there are resampled points along the segment P_iP_{i+1} as shown in Fig. 5.3. The first resampled point V_{i0} is designed to absorb the fragment $l_{E(i-1)}$ left by the previous segment. The distance between P_i and V_{i0} is denoted by l_{Bi}. The other resampled points are chosen to make the distance between adjacent points equal to l_{des}. At the end of the segment, there is usually a fragment l_{Ei}, less than l_{des}, left between the last resampled point and P_{i+1}.

The corresponding coordinates Vx_{ij} and Vy_{ij} of the j^{th} resampled point along the segment P_iP_{i+1} can be calculated by the following pseudo code as:

```
$l_i = \sqrt{(Px_i - Px_{i+1})^2 + (Py_i - Px_{i+1})^2}$
$l_{Bi} = l_{des} - l_{E(i-1)}$
$if(l_i \geq l_{Bi})$
            $n_i = trunc((l_i - l_{Bi})/l_{des})$
            total number of resampled points along $P_iP_{i+1}$ $= n_i + 1$
            for ( $j$ from 0 to $n_i$)
            $\frac{Vx_{ij}-Px_i}{Px_{i+1}-Px_i} = \frac{Vy_{ij}-Py_i}{Py_{i+1}-Py_i} = \frac{l_{Bi}+j*l_{des}}{l_i}$
        endfor
        $l_{Ei} = l_i - l_{Bi} - n_i \cdot l_{des}$
    endif
    elseif $(l_i < l_{Bi})$
        $n_i = -1$ so that the number of points $(n_i + 1) = 0$
        $l_{Ei} = l_i + l_{E(i-1)}$
    endelseif
```

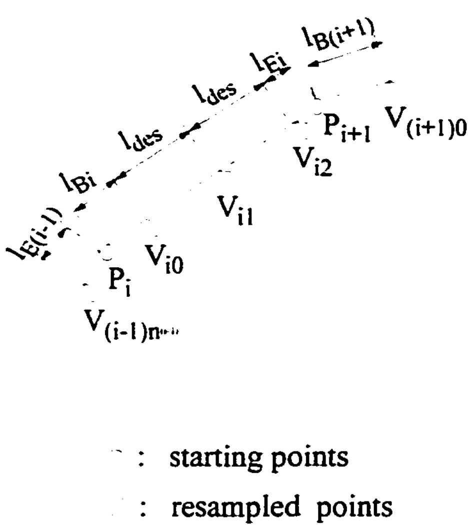


Figure 5.3. Illustration of the new contour resampling algorithm. The first resampled point V_{i0} is designed to absorb the fragment $l_{E(i-1)}$ left by the previous segment. $l_{E(i-1)} + l_{Bi} = l_{des}$. At the end of the segment there is another fragment l_{Ei}. $l_{Ei} = P_i P_{i+1} - l_{Bi} - 2 * l_{des}$. This fragment is suppose to be absorbed when the next segment is resampled.

Note that by this method, handling the last segment $P_{M-1}P_0$ may be different from other segments. That is the last fragment $l_{E(M-1)}$ can not be absorbed. If we just leave this fragment there, then the two resampled points V_{00} and $V_{(M-1)n_{M-1}}$ may cramp together when $l_{E(M-1)}$ is very small compared to the desired length l_{des}.

This problem can be suppressed by manipulating the last resampled point $V_{(M-1)n_{M-1}}$ according to two different conditions as:

- If $l_{E(M-1)} \leq \frac{1}{3}l_{des}$, then just simply drop the last point $V_{(M-1)n_{M-1}}$.
- If $l_{E(M-1)} > \frac{1}{3}l_{des}$, the last resampled point $V_{(M-1)n_{M-1}}$ is moved to the centre between the penultimate resampled point $V_{(M-1)(n_{M-1}-1)}$ and the first resampled point V_{00}, which is also the first starting point P_0. The coordinates of the adjusted last resampled point are given by:

$$Vx_{(M-1)n_{M-1}} = \frac{1}{2}(Vx_{(M-1)(n_{M-1}-1)} + Vx_{00})$$
$$Vy_{(M-1)n_{M-1}} = \frac{1}{2}(Vy_{(M-1)(n_{M-1}-1)} + Vy_{00})$$

We have gone through the main idea of our resampling method in the above discussion. Let's turn to a detailed steps involved in this method.

1. Input M, number of starting points, the coordinates of all starting points: $(Px_i, Py_i), i = 1, ..., M-1$, and the desired length between two resampled points: l_{des}.
2. Begin with i = 0 and set $l_{E(-1)}$, the fragment left by the last segment, to zero.
3. Calculate the total number of the resampled points along the segment P_iP_{i+1}, the coordinates of the resampled points, and the fragment l_{Ei} left.
4. check if $i > M - 1$ (if all the segments are handled).
    - if NO, go back to step 3.
    - if YES, adjust the last resampled point and stop the program.

As shown in the block diagram of the snake implementation in Fig. 5.1, this resampling algorithm is used at the beginning of each new iteration in the snake algorithm because:

- At the first iteration the starting points, often obtained manually, are likely to be randomly spaced.
- The points obtained by solving the linear equation system at the end of each iteration may cluster into a dense structure at certain places where there is a low energy field and the snake may fail to work on small objects with low resolution. Thus the resampling algorithm is applied after each iteration to ensure the points for next iteration are evenly spaced.

5.6 Determination of Elasticity and Rigidity Coefficients

As discussed in section 5.2 of the snake algorithm, we need to use a elasticity coefficient α_i and a rigidity coefficient β_i to compose matrix $\tilde{\mathbf{A}}$. During this project, we found α_i and β_i had great importance for the behavior of the snakes when the iterations continued as was also indicated in the literature [Lobregt and Viergever 1995; Kass et al. 1988; Cohen 1991; Ranganath 1995; Wang et al. 1996; Amini et al. 1990; Staib and Duncan 1992]. If α_i and β_i were very large, the internal energy had a major influence and the image forces had small effect. This resulted in the curve's shrinking to a point. On the contrary, if α_i and β_i were too small, the image energy had a major influence and the smoothness constraints had too small effect.

Unfortunately we have not found any theoretical formula for determining appropriate snake coefficients in the literature. In this project we experimentally found a good way to choose these coefficients is to make the entries of matrix $\tilde{\mathbf{A}}$ all have the same order of magnitude. For example, if step size τ was chosen as 1 as required for the stability of the snakes (see later), then choosing the coefficients as constant $\alpha_i = 0.5 - 1.5$ and $\beta_i = 0.5 - 1.5$ worked well.

5.7 Image Force

As discussed in section 5.2, step 6 of the snake algorithm requires calculation of the image force vector F. The image energy used in this project was the sum of the absolute Sobel gradients in all compass directions as:

$$E_{img}(V_x, V_y) = -|\bigtriangledown I(V_x, V_y)| = -\sum_{i=1}^{8} |Sobel_i(I(V_x, V_y))| \cdot w_{sb}(i)$$

where $w_{sb}(i)$ was the adjusting weight for the i^{th} Sobel mask operator.

The image force corresponding to the image energy in discretized form is:

$$\begin{aligned} F_x = \frac{\partial E_{img}}{\partial V_{x_i}} &= \frac{1}{2}(E_{img}(V_{x_i} + 1, V_{y_i}) - E_{img}(V_{x_i} - 1, V_{y_i})) \\ F_y = \frac{\partial E_{img}}{\partial V_{y_i}} &= \frac{1}{2}(E_{img}(V_{x_i}, V_{y_i} + 1) - E_{img}(V_{x_i}, V_{y_i} - 1)) \end{aligned} \qquad (5.1)$$

5.8 Stability of Snake Contour

One important concern about the implemented snakes is stability. Will the snake be locked to the desired boundary accurately without missing some boundary and without being trapped by noise?

Our experiments in this project agreed with Cohen's [1991] discussions. Even when an initial guess is close to an edge, it may not be led to the edge during the iteration. According to Equation 4.8, the position, $\mathbf{V}_n$, at n^{th} iteration is obtained after moving $\mathbf{V}_{n-1}$ along vector $\tau\mathbf{F}(\mathbf{V}_{n-1})$ and then solving the system, effectively smoothing the curve. If $\tau\mathbf{F}(\mathbf{V}_{n-1})$ is too large the point $\mathbf{V}_{n-1}$ can be moved far from the desired minimum and never come back. Thus the curve can pass through the edge and then make large oscillations.

To solve this instability, Cohen [1991] suggested to normalize the image force by taking $\mathbf{F}_{norm} = -\gamma(\mathbf{F}/||\mathbf{F}||)$, where γ is a weight coefficient. When $\tau\gamma$ is chosen as on the order of one pixel, $\tau\mathbf{F}_{norm}$ is about one pixel. Thus when a point is close to an edge point, the point will move about one pixel at a time and will not miss the edge. This implies the curve is attracted to the edge and stabilizes.

In this project the snake's instability of missing some boundary and being trapped by noise was reduced significantly by normalizing the magnitude of image force and making $\tau\gamma = 1$ as predicted by Cohen [1991].

5.9 Contour Detection for the Abdomen Image by Using the Snake

So far, we have discussed our implementation and considerations for the snakes in a theoretical manner. Now we apply this implementation of the snakes to the motion corrupted abdomen image as shown in Fig. 2.5. As we already showed in the last chapter the conventional Sobel mask operator failed to extract a continuous contour for this image. In this section, we show that our snake implementation produces a much better contour of the central image for this abdomen image. Fig. 5.4(a) shows the manually obtained points in the vicinity of the central image at the beginning of the snake algorithm. Fig. 5.4(b) shows the corresponding resampled points. Fig. 5.4(c)(d)(e) and (f) show the snaked contours after 20. 60. 100 and 120 iterations respectively. The used snake coefficients were: $\alpha = 0.5, \beta = 1.0.\ \tau = 1.0.\ \gamma = 1.0.$ The process of the snake evolution is very obvious from Fig. 5.4(b)-(f). The seed contour in Fig. 5.4(b) at the very beginning of the snake is far away from the desired contour. After 120 iterations, the snake is gradually attracted to the desired contour even at places where edges are very weak. Note that as shown in Fig. 5.4(f). the points of the snaked contour are not exactly at the sharpest edges everywhere along the contour. That is because the energy minimization of the snake is global. but not local. As is shown later in this thesis this will not affect the success of our composite method as long as the contour is continuous and close enough to the sharpest edges. Fig. 5.5 shows the initial contour (left) and the snaked contour (right) after 10 iterations for the phantom image.

5.10 Contour Ballooning and Its Applications for Obtaining Ghost Contours

As shown in Fig. 2.5, it is clear that contours of ghosts will be difficult to achieve by using directly the information displaying in the corrupted image as ghosts are very weak and smeared. To overcome this difficulty, contours of ghosts are considered as

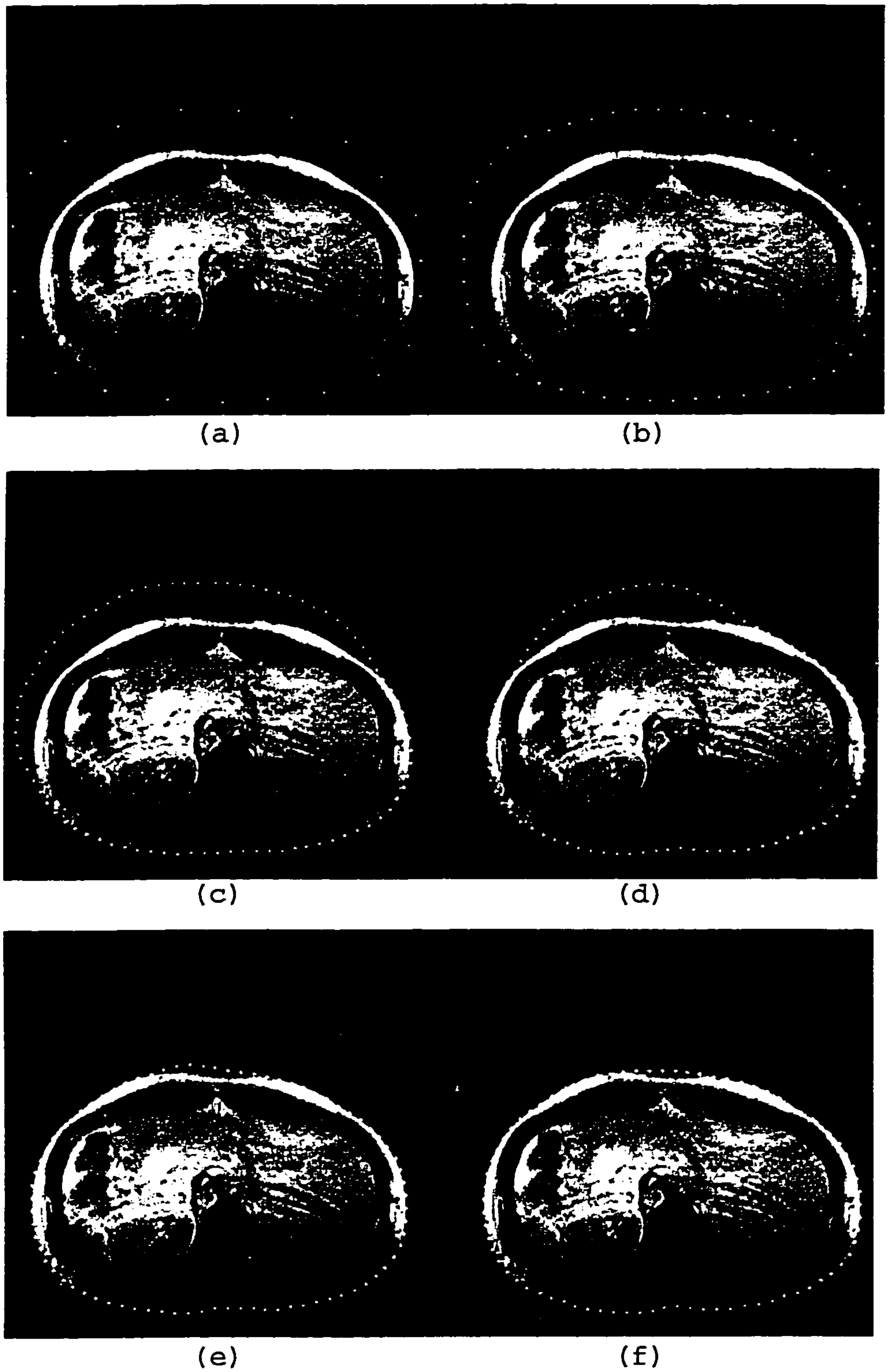


Figure 5.4. The results of snakes for the abdomen image. (a) Manually obtained points. (b) The resampled points of (a) used as seed contour for the snake. (c) Snake contour after 20 iterations. (d) Snake contour after 60 iterations. (e) Snake contour after 100 iterations. (f) Snake contour after 120 iterations.

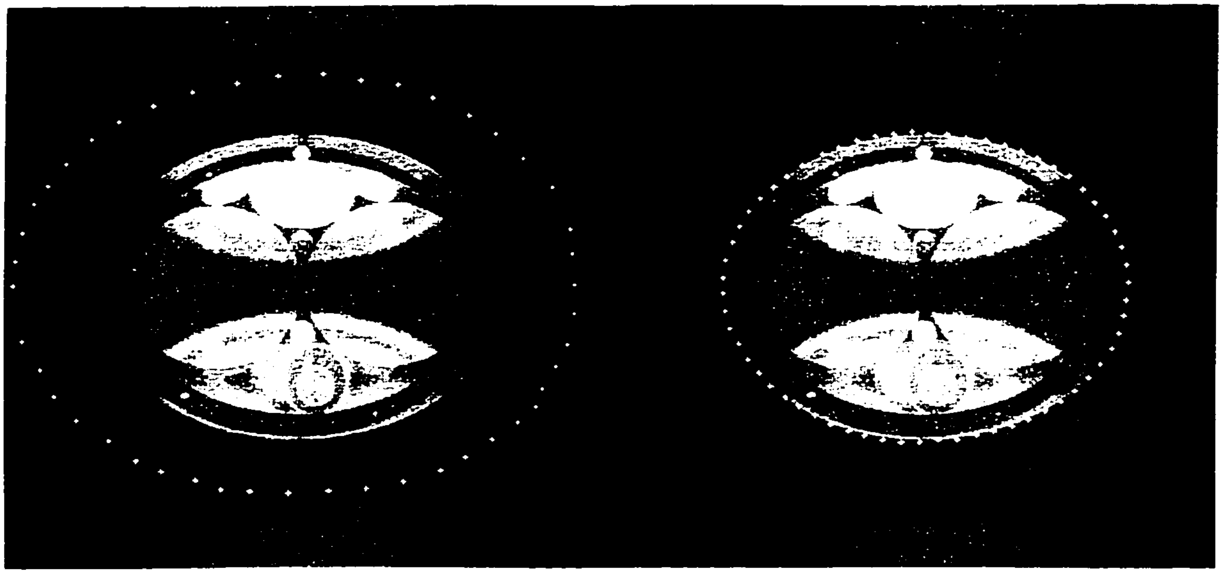


Figure 5.5. The result of snakes for the phantom image. The manually obtained points (left) are far away from the boundary detected by the snake algorithm (after 10 iterations) (right).

shifted and enlarged versions of that of the central image in this project as ghosts are often smeared and possibly enlarged. We proposed a contour ballooning method, which can be used to calculate any contour's enlarged versions.

Let's discuss this contour ballooning method. By ballooning contours, we mean points along original contours are moved a certain distance Δ in the direction of the normals at locations of these points. Movement toward the inside part of a contour is called deflating. Movement toward the outside part of a contour is called inflating.

Assume points along an original contour are $P_0, P_1, ..., P_{M-1}$ and the corresponding ballooned points are $B_0, B_1, ..., B_{M-1}$. First of all we need to determine the normals. The normal at point P_i is considered to be the normal of the line defined by points P_{i-1} and P_{i+1}. Thus B_i can be determined by:

For inflating:

$$\begin{aligned} Bx_i =& \ (Px_{i-1} + Px_{i+1})/2 - t(Py_{i+1} - Py_{i-1}) \\ By_i =& \ (Py_{i-1} + Py_{i+1})/2 + t(Px_{i+1} - Px_{i-1}) \end{aligned}$$

For deflating:

$$\begin{aligned} Bx_i =& \ (Px_{i-1} + Px_{i+1})/2 + t(Py_{i+1} - Py_{i-1}) \\ By_i =& \ (Py_{i-1} + Py_{i+1})/2 - t(Px_{i+1} - Px_{i-1}) \end{aligned}$$

where $t = \Delta/\sqrt{(Px_{i-1} - Px_{i+1})^2 + (Py_{i-1} - y_{i+1})^2}$, Δ is the ballooning distance from P_i to B_i.

Fig. 5.6 shows the shifted and ballooned (Δ=5 pixels) versions of Fig. 5.4(f) for the upper and lower ghosts in the abdomen image. Using this contour approach, it is straightforward to account for the ghost image aliasing that occurs when a ghost's shifted contours appear in both the upper and lower portions of the image.

5.11 Summary

In this chapter, our implementations of the snakes, based on the discretized snake model and the evolution solution of a linear equation system, is discussed in detail. The issues associated with our implementation of the snakes include:

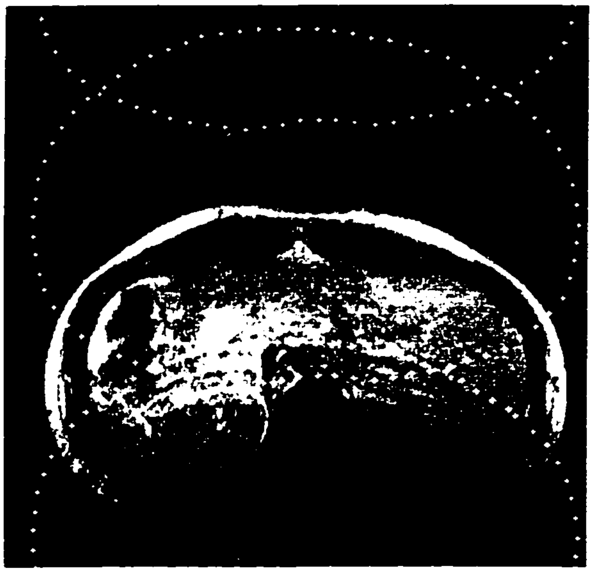


Figure 5.6. Contours for the upper and lower ghosts in the abdomen image obtained by ballooning and shifting the snaked contour of the central image. Note the aliased lower ghost image.

- To attract snakes far away from the desired edges, a Gaussian low-pass filter was used to blur edges. By repeatedly running the snake algorithms more than once with reduced blurring of edges, snake contours could be gradually locked to the desired edges.

- To ensure points along a contour were evenly spaced for the snakes, a new resampling algorithm was proposed.

- The selection of elasticity coefficient and rigidity coefficient had great influence on the behavior of snakes.

- The normalized image force gave better stability to snakes.

Finally it is shown that by using our implementation of the snakes. continuous contours are achieved for the central images of both the abdomen and phantom images. Contours of the ghost images are considered as ballooned versions of the corresponding central images.

In the next chapter we will discuss how to use the snake contours of the central and ghost images to segment the corrupted image prior to determining of the distortion transfer function (DTF) and the composite method of motion artifact suppression. We shall show how to take the snake contour information and. using a new approach. segment the image.

CHAPTER 6

CONTOUR-BASED REGION LABELING

6.1 Introduction

In the papers describing the snakes the authors normally are satisfied with knowledge of the contour. In this thesis we need to proceed further and segment the image into various regions. In this chapter we show how to use the contour information (discussed in previous two chapters) to identify various segments of an image data matrix prior to using this information in determining the distortion transfer function (DTF) and correcting motion artifacts. We will provide details of how we modified a computer graphics scan-conversion method to label image regions given known contours. For this approach to be successful, a complete (full) set of points on the contour is required. We show how to use a modified B-spline algorithm to guarantee this full set of points. The use of this approach is demonstrated by labeling the various regions of a motion corrupted MRI image.

6.2 B-spline Curve Fitting

It is well known in computer graphics that unless a full set of contour points is known, scan-conversion algorithms will self-destruct. This problem is also present in this MRI application. This instability is overcome by applying B-spline to the snake contour information.

As discussed in the last chapter, what we obtain after applying the snake algorithms is M discrete points that lie on the desired boundary of an object. However normally a few of discrete points do not represent a contour completely and uniquely. To describe a contour completely, a continuous curve representation is required. The task of getting a continuous curve representation from known points is called curve fitting. In this project, a so-called B-spline curve fitting method is employed. The discrete contour points obtained by the snake algorithms were sent to the B-spline algorithms as control points in order to get a continuous representation for the contour.

B-spline is a very popular method for curve fitting and interpolating. There are superior properties for B-spline [Foley et al. 1996; Dierckx 1982], such as:

- B-splines have good continuity.
- B-splines' coefficients depend on just several control points. This property is called local control property of B-spline.

With B-splines, an entire curve consists of several curve segments. Each curve segment is expressed by a weighted sum of polynomial basis functions. In this project, the uniform cubic B-splines are used. By the cubic B-spline, not only the curve itself but also its first-degree and second-degree derivatives are continuous. Assume we know a set of $m+1$ control points, $C_0, C_1, C_2, ..., C_m, m > 3$, on a contour (for a closed curve the control points are easy to create by repeating the points C_0, C_1, C_2 at the end of the sequence so that the control points look like this: $C_0, C_1, C_2, ..., C_m, C_0, C_1, C_2$. A curve is represented by $m-2$ cubic polynomial curve segments $Q_3, Q_4, ..., Q_m$ when we use a uniform B-splines. Each cubic polynomial is completely defined by four adjacent points out of the $m+1$ control points. For example, Q_3 is defined by C_0, C_1, C_2 and C_3. While Q_4 is defined by C_1, C_2, C_3 and C_4. Generally $Q_i, 3 \leq i \leq m$, is defined by the four control points $C_{i-3}, C_{i-2}, C_{i-1}$ and C_i. Thus, the cubic B-spline geometry vector $G_{B_{s_i}}$ for segment Q_i is defined as:

$$\mathbf{G}_{B_{s_i}} = \begin{bmatrix} C_{i-3} \\ C_{i-2} \\ C_{i-1} \\ C_i \end{bmatrix}, 3 \leq i \leq m.$$

If we define $\mathbf{T}_i$ as the row vector $[t^3 \;\; t^2 \;\; t \;\; 1]$, where t is the variable of these polynomial and $t \in [0, 1)$, then the cubic B-spline formulation for curve segment Q_i is :

$$Q_i(t) = \mathbf{T}_i \cdot \mathbf{M}_{B_s} \cdot \mathbf{G}_{B_{s_i}}, \quad 0 \leq t < 1 \tag{6.1}$$

where $\mathbf{M}_{B_s}$ is called B-spline basis matrix. The polynomial coefficients for each curve segment are exactly the same for uniform cubic B-splines:

$$\mathbf{M}_{B_s} = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

The so called B-spline blending functions $\mathbf{B}_{B_s}$ are given by the product of $\mathbf{T}_i \cdot \mathbf{M}_{B_s}$. The blending functions for each curve segment are also exactly the same for uniform cubic B-splines:

$$\begin{aligned} \mathbf{B}_{B_s} &= \mathbf{T} \cdot \mathbf{M}_{B_s} = [B_{B_{s_{-3}}} \quad B_{B_{s_{-2}}} \quad B_{B_{s_{-1}}} \quad B_{B_{s_0}}] \\ &= \frac{1}{6}[-t^3 + 3t^2 - 3t + 1 \quad 3t^3 - 6t^2 + 4 \quad -3t^3 + 3t^2 + 3t + 1 \quad t^3] \\ &= \frac{1}{6}[(1-t)^3 \quad 3t^3 - 6t^2 + 4 \quad -3t^3 + 3t^2 + 3t + 1 \quad t^3], \quad 0 \leq t < 1. \end{aligned}$$

Thus the closed continuous contour has a complete representation as segments Q_3, Q_4, ..., Q_m, each of which is represented by a continuous polynomial as in Equation 6.1. The curve is continuous not only within each segment but also at the joint points between segments. For example, at the joint point between segment Q_i and segment Q_{i+1}, $3 \leq i \leq m$, we have $Q_i(1) = Q_{i+1}(0)$, $Q_i'(1) = Q_{i+1}'(0)$ and $Q_i''(1) = Q_{i+1}''(0)$. Joint points at $Q_i(1) = Q_{i+1}(0)$ are called knots.

As Cheney and Kincaid [1994] stated, in general, the fitted curve does not have to pass through the control points. Curve segment Q_i begins somewhere near control point C_{i-2} and ends somewhere near control point C_{i-1}. The B-spline blending functions are everywhere nonnegative and sum to unity, so the curve segment Q_i is constrained to the convex hull of its four control points. Just as each curve segment is defined by four control points, each control point influences four curve segments (except for those at the beginning and end of the sequence $C_0, C_1, \ldots, C_m$ in the case of nonclosed curves). Moving a control point in a given direction moves the four curve segments it affects in the same direction. The other curve segments are totally unaffected. This is the local control property of B-splines.

6.3 A Full Set of Points of a Contour

As discussed in the above section, a continuous expression can be obtained from M discrete points along a contour by using the B-spline curve fitting algorithm. The question discussed in this section is how to make use of this continuous expression to represent a closed contour in digital form completely and uniquely. In order for a digital scan conversion algorithm to work, a full (complete) set of points is required. By a full set of points, we mean all the points where the integer grid lines (pixel locations) intersect with the continuous curve. This indicates the horizontal coordinate distance and vertical coordinate distance between any two adjacent points P_j and P_{j+1} in a full set of a contour should not be greater than one pixel. In other words, only the following three cases are allowed:

$$case1 : \begin{cases} |P_{x_j} - P_{x_{j-1}}| = 0; \\ |P_{y_j} - P_{y_{j-1}}| = 1; \end{cases}$$

$$case2 : \begin{cases} |P_{x_j} - P_{x_{j-1}}| = 1; \\ |P_{y_j} - P_{y_{j-1}}| = 0; \end{cases}$$

$$case3 : \begin{cases} |P_{x_j} - P_{x_{j-1}}| = 1; \\ |P_{y_j} - P_{y_{j-1}}| = 1; \end{cases} \tag{6.2}$$

If we were dealing with a simple line or an ellipse shaped object, then an investigation of the Bresenham algorithm for line drawing would be appropriate [Foley et al. 1996]. However in this thesis we are considering a line drawn to a general cubic curve which is not handled by a Bresenham algorithm. In addition since these calculations are performed offline, a highly efficient algorithm is not necessary.

In this project, a full set of points is obtained by giving parameter t appropriate values. We developed a check-and-insert algorithm for this purpose. The algorithm works as follows:

1. Start with the first polynomial curve segment Q_3 by setting $i = 3$ and reset the sequence number, j, for the inserted points along this segment to zero.

2. Stop if $i > M - 1$, meaning all the polynomial curve segments are handled.

3. Calculate Δ_i, the bigger of the horizontal and vertical coordinate distances between $Q_i(0)$ and $Q_i(1)$. The default step size Δt_i for this segment Q_i is given by $\Delta t_i = 1/\Delta D_i$.

4. Set $t = 0$, $P_j = Q_i(0)$ and $\Delta t = \Delta t_i$ at the beginning of a new segment Q_i.

5. Increase t by Δt.

6. Go back to step 2 with $i = i + 1$ for the next segment if $t \geq 1$, meaning the segment Q_i is complete.

7. Calculate a potential inserted point using $\tilde{P} = Q_i(t)$.

8. Check if Equation 6.2 is satisfied or not.

   - If YES, insert the point $\tilde{P}$ as the next point along the curve by setting $j = j + 1$ and $P_j = \tilde{P}$. Set $\Delta t = \Delta t_i$ and go back to step 5 to search for the next point.

   - If NO, meaning Δt is still too big for calculating the next point. Calculate $\Delta\tilde{D}$, the larger of the horizontal and vertical coordinate distance between the potential point $\tilde{P}$ and P_j.

9. Draw t back by subtracting Δt. Set the new step size (smaller) $\Delta t = \Delta t/\Delta\tilde{D}$. Increase t by Δt and go back to step 6 for a new try.

We applied this check-and-insert algorithm to the contour points of the abdomen image obtained by the snakes in the last section. Fig. 6.1(a) shows the contour points obtained by the snakes. Fig. 6.1(b) shows its fully interpolated contour by this check-and-insert algorithm. Results in Fig. 6.1 indicate our approach of obtaining a full set of points is successful.

Note that by the check-and-insert algorithm some interpolated points obtained by using incremented t values may be rounded to the same integer points. This is not going to be a problem as it makes no difference if a contour point is labeled once or twice or more times in our region labeling algorithm discussed below.

6.4 Region Labeling Algorithm

Now that a full set of points on the boundary of a region has been determined. the next step is to decide which pixels are inside the region and label them with a flag value in order to form the region. In this project a fast region labeling algorithm based on a computer graphics generalized scan-conversion algorithm is proposed. The foundation of this generalized scan-conversion algorithm is the odd-parity rule in computer graphics [Foley et al. 1996]. The odd-parity rule says: to determine whether a region lies inside or outside a given region, choose as a test point any point inside the particular region. Next, choose a ray that starts at the test point and extends infinitely in any direction. If this ray intersects the region outline an odd number of times, the region is considered to be interior. If the ray intersects an even number of times, the region is considered to be outside the contour. Our generalized scan-conversion algorithm does not perform the odd-parity test directly. Instead, the odd-parity rule is efficiently applied to an entire row of adjacent pixels that lie either inside or outside the contour.

In scan-conversion techniques, the scan lines are of 1-pixel width. Without loss of generality, let's assume the scan lines run from left to right in horizontal direction

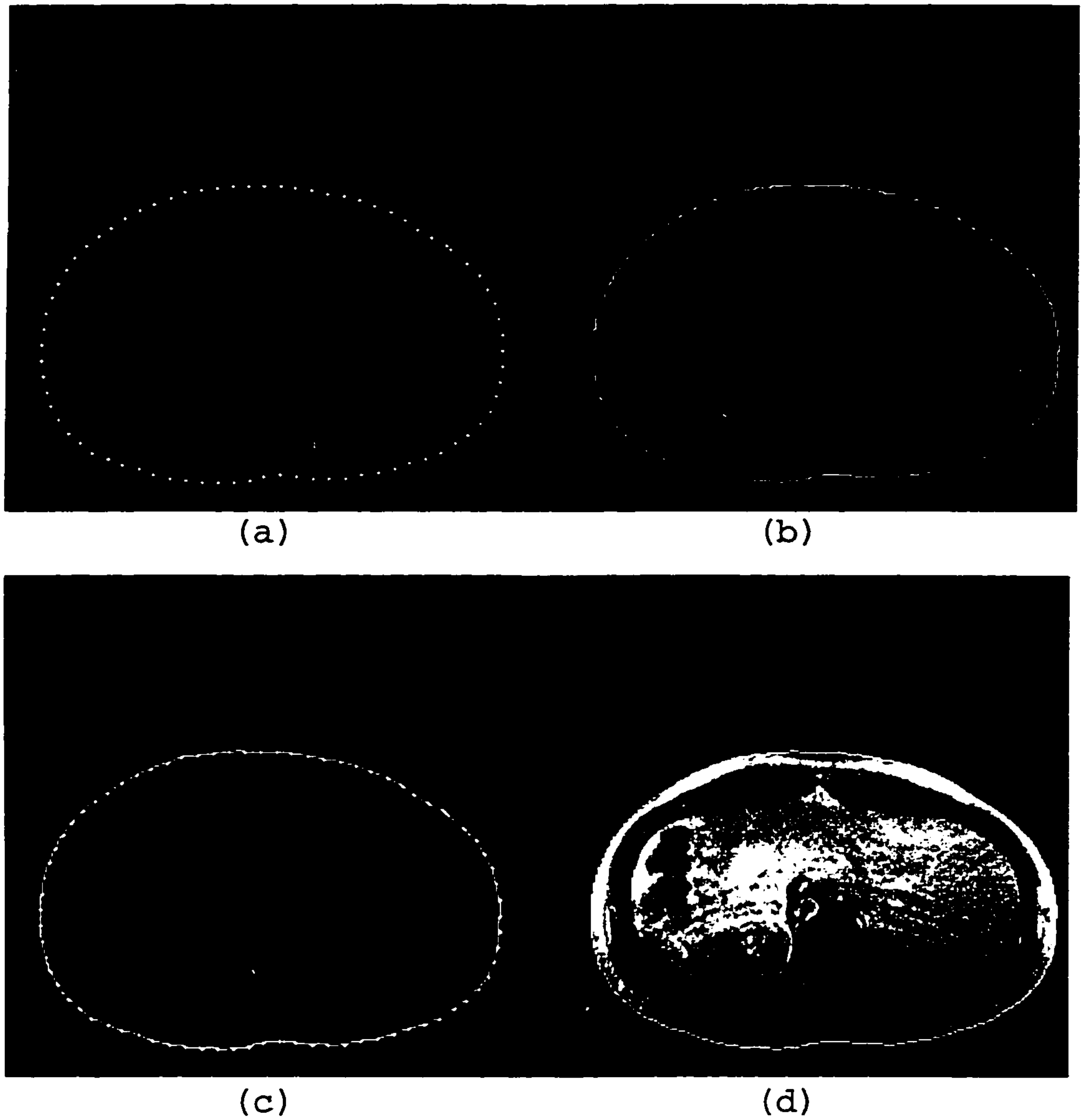


Figure 6.1. Results of the check-and-insert algorithm. (a) Points along the snake contour. (b) A full set of points obtained by using the check-and-insert algorithm. (c) The superposition of the snake contour and the result of the check-and-insert algorithm (d) The full set of points with the abdomen image as background.

(the x direction). The scanning is taken successively from the very top of the contour image to the very bottom of the contour image. For each scan line the intersected points between the scan line and the contour from left to right are recorded. If the intersection is a single point, it is treated as a valid intersection. The corresponding x coordinate is considered as the span extremum and the parity is converted. At the beginning of each scan line, we are outside the region of support of the MRI central image. There is no intersection and the parity is set to even. After a valid intersection, the parity is converted to odd, which means a span inside the contour begins and the left span extremum x_{begin} is recorded. If the next intersection is also a single point, the parity is converted from odd back to even, which means this span inside the contour ends and the right span extremum x_{end} is recorded. The span from x_{begin} to x_{end} is labeled as interior and the searching for a new interior span begins. Normally for a closed contour, the number of intersections is even. In other words, the intersections appear as pairs.

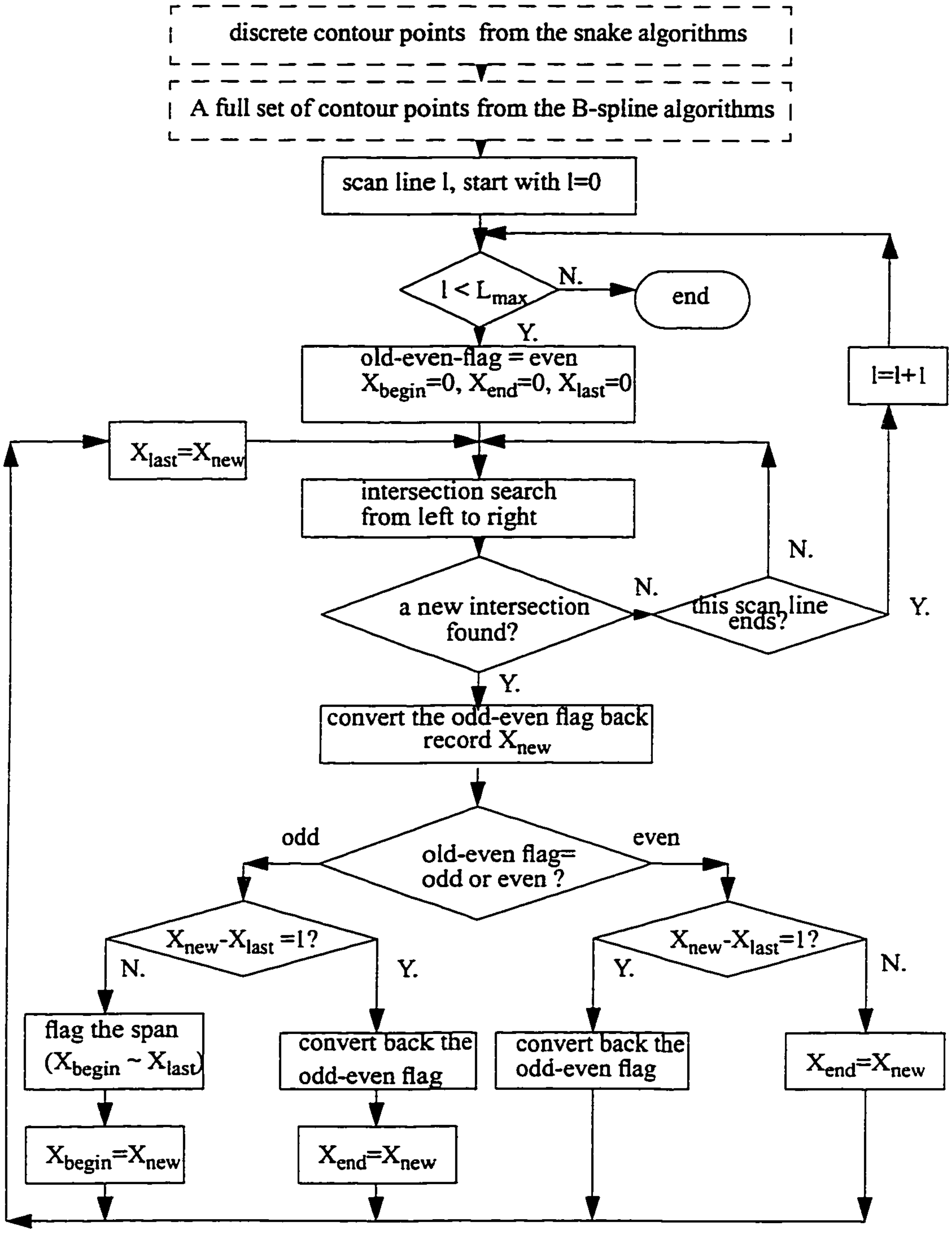
However the situation get more complicated in practice when the contours are much more complicated than polygons, normally considered in computer graphics applications. One problem we met in our experiments is where the contour segments are horizontal and parallel to the scan lines. If that happens, the intersected points are no long isolated points but many points clustered together and the odd-parity rule fails to work. To overcome this difficulty, we generalize the way of defining valid intersections. A cluster of intersecting points, which must be a segment of contour, is treated as one single valid intersection for the odd-parity rule. By a cluster of intersections, we mean the horizontal coordinate distance between the adjacent intersected points is 1-pixel. The block diagram of this generalized scan-conversion algorithm is shown in detail in Fig. 6.2. The top two boxes with dotted lines in Fig. 6.2 show that in this project the input of the generalized scan-conversion algorithm is from the output of the B-spline curve fitting algorithms, whose input is from the output of the snake algorithms.

The generalized scan-conversion algorithm works well in this project. Fig. 6.3 gives the labeling result of the central image's contour in the abdomen image. As we will see later, this labeled region is treated as the region of support of the central image and notated as R_c in Chapter 8 when the composite method is discussed.

However we also notice that this generalized scan-conversion algorithm does not fit for all cases mathematically as the odd-parity rule may fail at some special situations. Fig. 6.4 shows examples that the odd-parity rule can not handle appropriately. Although all the three contours have three valid intersections at the scan line, the interior spans are different for these three contours. Fortunately that does not happen often. Most of our practical cases were not too complicated for the scan-conversion algorithm.

6.5 Summary

Given some points on a contour, the B-spline curve fitting algorithm can be applied to get a complete (full) set of points along the contour. Based on the full set of points, the region inside the contour is labeled using our new segmentation approach based on a generalized scan-conversion algorithm. The odd-parity rule in computer graphics is the foundation of this generalized scan-conversion method. In the next two chapters this region labeling method is used for determining distortion transfer function (DTF) and during the suppression of motion artifacts.


discrete contour points from the snake algorithms
A full set of contour points from the B-spline algorithms
scan line l, start with l=0
l < L_{max}
N.
end
Y.
old-even-flag = even
X_{begin}=0, X_{end}=0, X_{last}=0
l=l+1
X_{last}=X_{new}
intersection search
from left to right
N.
a new intersection found?
N.
this scan line ends?
Y.
Y.
convert the odd-even flag back
record X_{new}
odd
old-even flag= odd or even ?
even
X_{new}-X_{last} =1?
N.
Y.
X_{new}-X_{last}=1?
Y.
N.
flag the span (X_{begin} ~ X_{last})
convert back the odd-even flag
convert back the odd-even flag
X_{end}=X_{new}
X_{begin}=X_{new}
X_{end}=X_{new}

L_{max}:	the number of scan lines.
X_{last}, X_{new}:	the x coordinates of the last and this intersections respectively
X_{begin}, ,X_{end}:	the left and right span extremums respectively

Figure 6.2. The block diagram of the generalized scan-conversion algorithm. The top two boxes with dotted lines show that the input of this algorithm is from the output of the B-spline curve fitting algorithms, whose input is from the output of the snake algorithms.

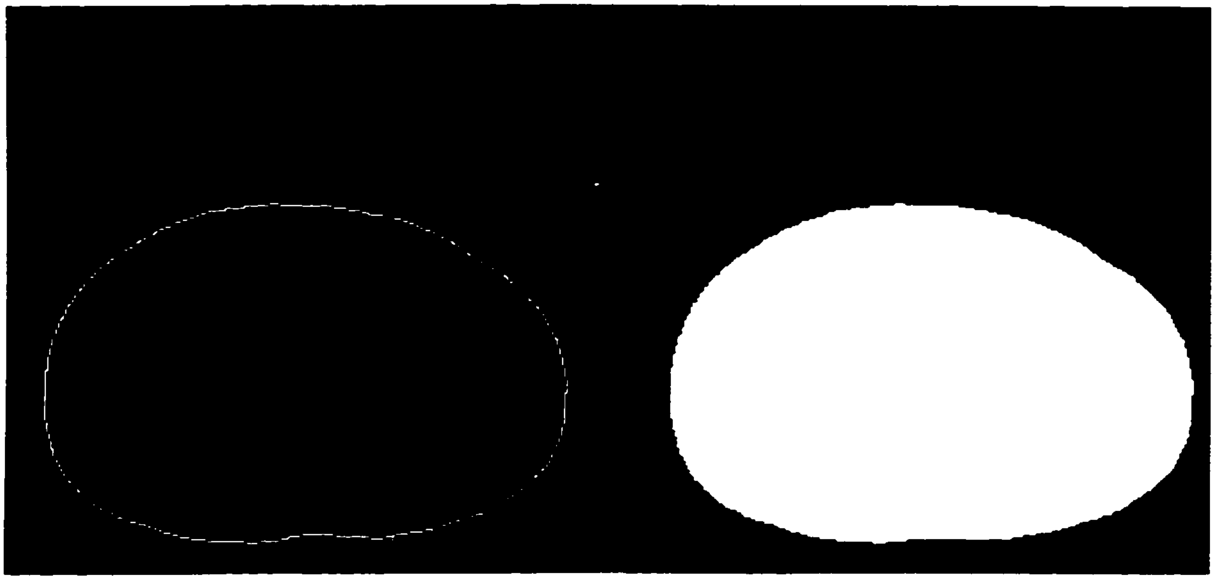


Figure 6.3. Region labeling result (right) of the full set of points (left)

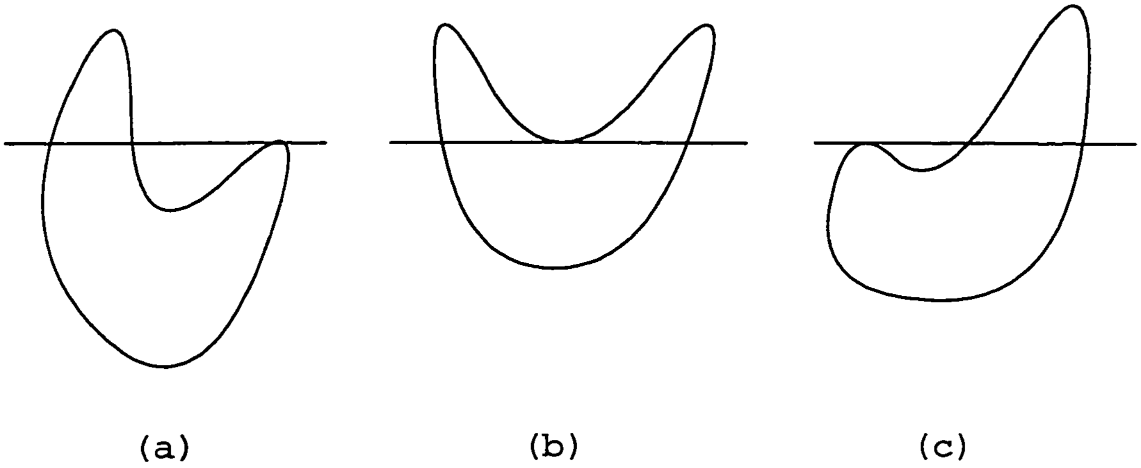


Figure 6.4. Too complex situations for the odd-parity rule to work

CHAPTER 7

DISTORTION TRANSFER FUNCTION ESTIMATION

7.1 Introduction

In this chapter, a method of estimating distortion transfer functions (DTFs) directly from motion corrupted images is proposed and investigated. The results of applying this approach to phantom images corrupted by a variety of DTFs are given. We describe our observations on what limits the accuracy of the DTF estimation. Finally DTFs estimated by using this approach for the abdominal image with unknown respiratory motion are given.

A paper based on this chapter has been submitted to IEEE Trans. on Medical Imaging. The article is entitled "Using a MRI Distortion Transfer Function to Characterize Motion of a Non-rigid Object" by Yang and Smith [1998].

7.2 Estimation of Distortion Transfer Function for Non-rigid Motion

The problem to be investigated in this section is: how to determine all the parameters and DTFs of all the ghosts given a corrupted MR image? There are three parameters that need to be determined (Equation 3.5) in order to identify the characteristics of ghosts:

- N, the number of ghosts in the corrupted image.

- Δ_n, $n = 0, 1, ..., N-1$: the distance between the n^{th} ghost image and the central image.

- $DTF_n(k_x, k_y)$, $n = 0, 1, ..., N-1$: the distortion transfer function of each ghost image.

In this project, N_{major}, the number of major ghosts, and $\Delta_n, n = 1, 2, ... N_{major}$, distance between the n^{th} ghost and the central image, were manually estimated directly from the corrupted image. Zeng and Smith [1994] reported a procedure to analyze the position of the ghosts by characterizing the modulation in the project k-space data. However this automated approach typically fails when the images are smeared because of the non-rigid motion.

A major difficulty lies in providing an estimate of the DTF_n for each ghost image. Mathematically, the distortion transfer function is from:

$$DTF_n(k_x, k_y) = F_{g_n-centred}(k_x, k_y)/F_{no-motion}(k_x, k_y) \qquad (7.1)$$

where $F_{g_n-centred}(k_x, k_y)$ and $F_{no-motion}(k_x, k_y)$ are the Fourier transform of the n^{th} ghost image shifted Δ_n to the position of the central image and the central (correct) image respectively. By moving the ghost image to the position of the central image. we are removing Mitsa's phase factor in the image domain.

However the typical overlapping of the central and ghost images means that the calculation of $DTF_n(k_x, k_y)$ can not be performed directly. We use Fourier transforms of the unoverlapped parts of the ghost and central images instead of that of the whole ghost and central images to avoid performing the above operation (Equation 7.1). Using Fourier transforms of the unoverlapped part of the n^{th} ghost and the corresponding unoverlapped part of the correct image. an estimate of $DTF_n(k_x, k_y)$ is :

$$DTF_{est_n}(k_x, k_y) = \mathcal{F}(I_{g_n-unoverlap-centred}(x, y))/\mathcal{F}(I_{c-unoverlap}(x, y)) \qquad (7.2)$$

where $I_{g_n-unoverlap-centred}(x, y)$ is the unoverlapped parts of the n^{th} ghost shifted Δ_n back to the position of the central image. $I_{c-unoverlap}(x, y)$ is the corresponding unoverlapped parts of the central image.

There is resemblance between the estimation of the DTF represented by Equation 7.2 and the short time Fourier transform in the one-dimensional case in that they both use part of the signal to get frequency information when it is impossible to use the whole signal.

However we found that estimates of DTF by using Equation 7.2 directly are often not accurate enough. That is not hard to understand because the unoverlapped parts have the effect of imposing artificially sharp edges to the central and ghosts' images. These artificial edges generally introduce undesirable digital processing characteristics [Harris 1978] into the deconvolution in Equation 7.2. In the next section we discuss a windowing technique to reduce these undesirable characteristics caused by the artificial edges and increase the accuracy of the DTF estimation.

7.3 The Windowing Techniques for DTF Estimation

By applying a window onto the unoverlapped parts of the central and ghost image. Equation 7.2 can be approximated by:

$$D\tilde{T}F_{est_n}(k_x, k_y) = \frac{\mathcal{F}(I_{g_n-unoverlp-centred}(x,y)\cdot wnd_1(x,y))}{\mathcal{F}(I_{c-unoverlp}(x,y)\cdot wnd_1(x,y))} \qquad (7.3)$$

where $wnd_1(x,y)$ is a two-dimensional window function.

One-dimensional windowing techniques are discussed in detail by Harris [1978]. Normally the Hanning, Hamming and 3-term Blackman profiles are better than rectangular profiles in terms of a highly concentrated central lobe with very low sidelobe structure. We have generated suitable 2D windows. A complication in this application is choosing the best location for the window centre to make optimum use of the information in the image. We chose the size and position of the windows to maximize the area windowed but still avoid any overlapped portions in order to make full use of the unoverlapped parts. As shown in Fig. 7.1, the unoverlapped parts are of pie shaped for object such as the human abdomen whose cross section has a shape similar to an ellipse. Also because the edges of ghosts contain more comprehensive information about motion, the centres of the windows are chosen close to the edges but goes slightly inside the object. As to forming two-dimensional windows, we chose to make the 2D windows radially symmetric about their centres in this project. The reason is because which direction contains more useful information about the motion, and thus should be emphasized, varies from image to image.

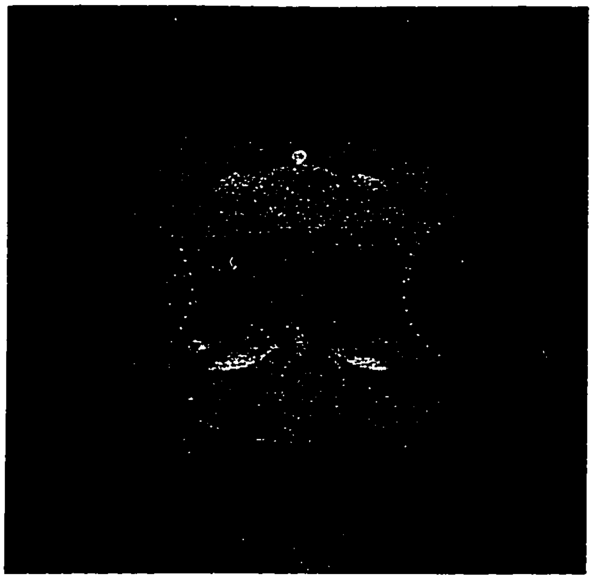


Figure 7.1. The window positions and sizes used to determine the DTFs of the motion corrupted image

The Fourier transform of the DTF represents how each individual image point is convoluted with a distortion point spread function. Simple application of $D\tilde{T}F_{est_n}(k_x, k_y)$ leads to a distortion point spread function that is unrealistically infinite in extension. This has the effect of enhancing noise induced instability. We reduced the effect of the instability by limiting the size of the distortion point spread function so that the estimated $DTF(k_x, k_y)$ was given by:

$$D\tilde{\tilde{T}}F_{est_n}(k_x, k_y) = \mathcal{F}[wnd_2(x, y) \cdot \mathcal{F}^{-1}(D\tilde{T}F_{est_n})] \qquad (7.4)$$

where $wnd_2(x, y)$ is the second window applied in the image domain.

In some situations, such as in presence of high noise, this additional windowing could make the ghost image lose intensity relative to the central image. In this situation, it would be necessary to introduce a compensation factor into the DTF to ensure the ghost intensities remain correct.

In the next section, we apply this DTF estimation method represented in Equation 7.4 to phantom images corrupted by a variety of DTFs.

7.4 The estimated DTFs for a variety of phantom images

In Chapter 3, we mentioned generation of a series of motion corrupted images using a variety of different DTFs. We applied Hanning windows to isolate selected portion of the ghosts and central image using the window centres and sizes (radius is about 30 pixels) as shown in Fig. 7.1. The second spatial domain window $wnd_2(x, y)$ used to suppress noise effects on the distortion point spread function was fairly wide with a radius of 25 pixels. The estimated DTFs for phantom images corrupted by a variety of DTFs are shown in Fig. 7.2 - Fig. 7.9.

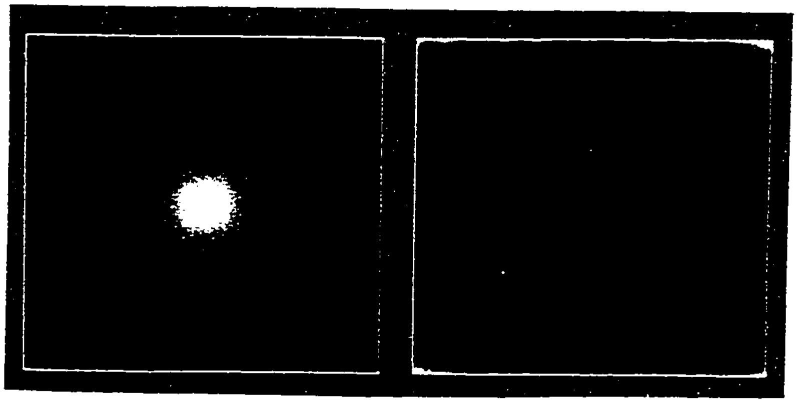


Figure 7.2. The estimated DTF (left) for the phantom image corrupted by DTFs having a form of Gaussian low-pass filter and the difference image between the theoretical and estimated DTF (right, intensity scaled up 3 times)

Fig. 7.2 - Fig. 7.9 show clearly that the estimated DTFs are rather accurate, which indicates our method for estimating DTFs directly from the motion corrupted images

Figure 7.3. The estimated DTF (left) for the phantom image corrupted by DTFs having a form of Butterworth low-pass filter and the difference image between the theoretical and estimated DTF (right, intensity scaled up 3 times)

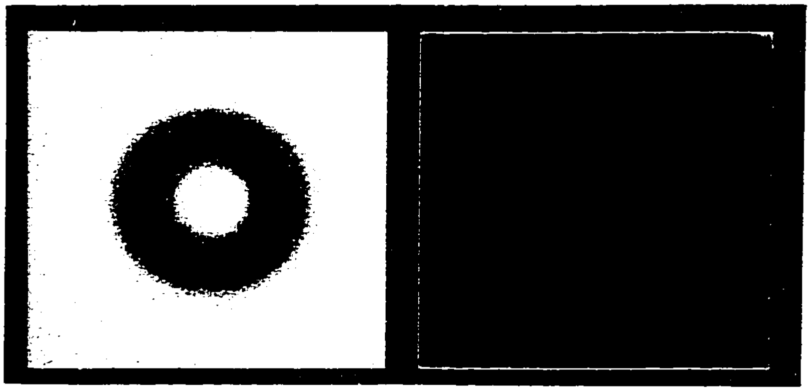


Figure 7.4. The estimated DTF (left) for the phantom image corrupted by DTFs having a form of radially symmetric Butterworth band-reject filter and the difference image between the theoretical and estimated DTF (right, intensity scaled up 3 times)

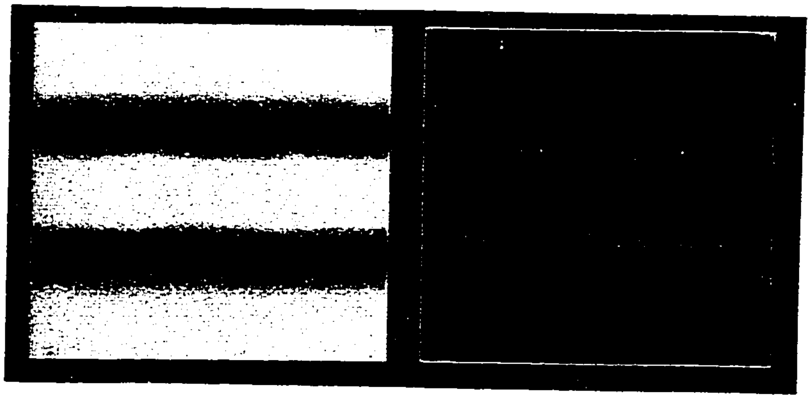


Figure 7.5. The estimated DTF (left) for the phantom image corrupted by DTFs having a form of belt Butterworth band-reject filter and the difference image between the theoretical and estimated DTF (right, intensity scaled up 3 times)

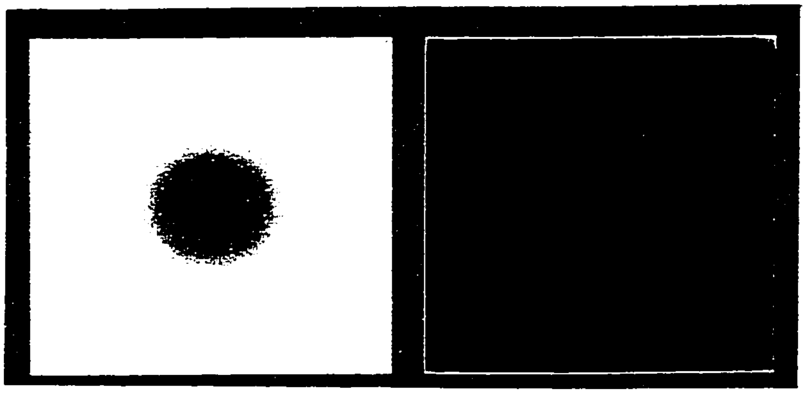


Figure 7.6. The estimated DTF (left) for the phantom image corrupted by DTFs having a form of Gaussian high-pass filter and the difference image between the theoretical and estimated DTF (right, intensity scaled up 3 times)

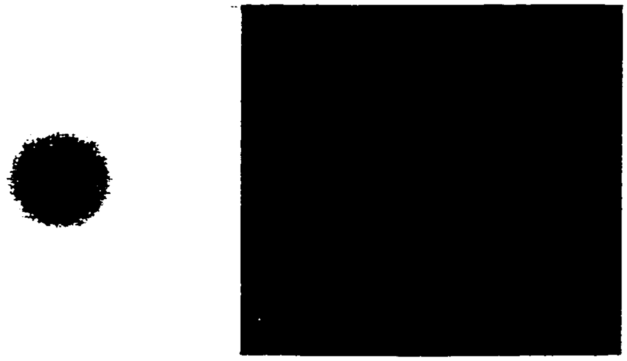


Figure 7.7. The estimated DTF (left) for the phantom image corrupted by DTFs having a form of Butterworth high-pass filter and the difference image between the theoretical and estimated DTF (right, intensity scaled up 3 times)

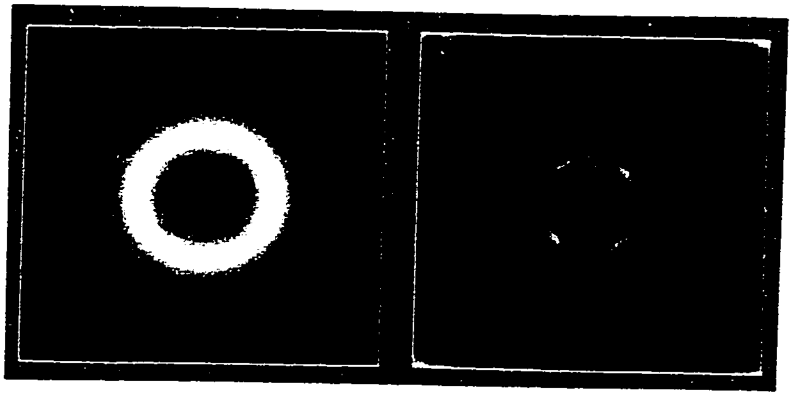


Figure 7.8. The estimated DTF (left) for the phantom image corrupted by DTFs having a form of radially symmetric Butterworth band-pass filter and the difference image between the theoretical and estimated DTF (right, intensity scaled up 3 times)

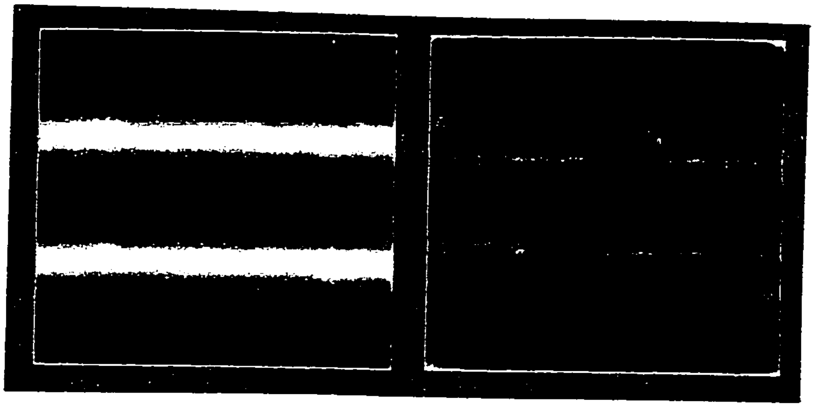


Figure 7.9. The estimated DTF (left) for the phantom image corrupted by DTFs having a form of belt Butterworth band-pass filter and the difference image between the theoretical and estimated DTF (right, intensity scaled up 3 times)

is feasible. It is also noticed that the estimation of DTFs having forms of low-pass filters is more accurate than others.

7.4.1 Observation: Image itself has effects on the accuracy of DTF estimation

Our experiments indicate that for accurate determination of the DTF there must be sufficient detail in the unoverlapped portions of the central and ghost images. Fig 7.10(upper) and (lower) show the experimentally determined DTFs for the phantom images corrupted by DTFs having a form of Gaussian low-pass filter and belt Butterworth band-pass filter respectively. The left hand side DTFs were estimated by using only left side unoverlapped part. While the right hand side DTFs were estimated by using only the right side unoverlapped part. From Fig. 7.10 it is clear that the estimated DTFs from the left side unoverlapped parts are more accurate than that from right side overlapped parts, which suggests the accuracy of DTF estimation is related to whether there was high detail (left) or low detail (right) present in the portion of the central image uncorrupted by motion artifacts. The ideal, but not practical, case is that the unoverlapped part only consists of an isolated point in a homogeneous background. In this case the distortion of the isolated point will accurately describe the distortion in all directions associated with the DTF.

7.5 The estimated DTFs for the abdomen image

For real MRI images, usually nothing is known about the motion except the corrupted images themselves. From the abdomen image itself as shown in Fig. 2.5, the parameters about the ghosts were estimated as:

- $N = 2$ as two major ghosts can be seen clearly in Fig. 2.5.
- As mention earlier, the automated approach for analyzing the positions of ghosts proposed by Zeng and Smith [1994] fails when the images are smeared because

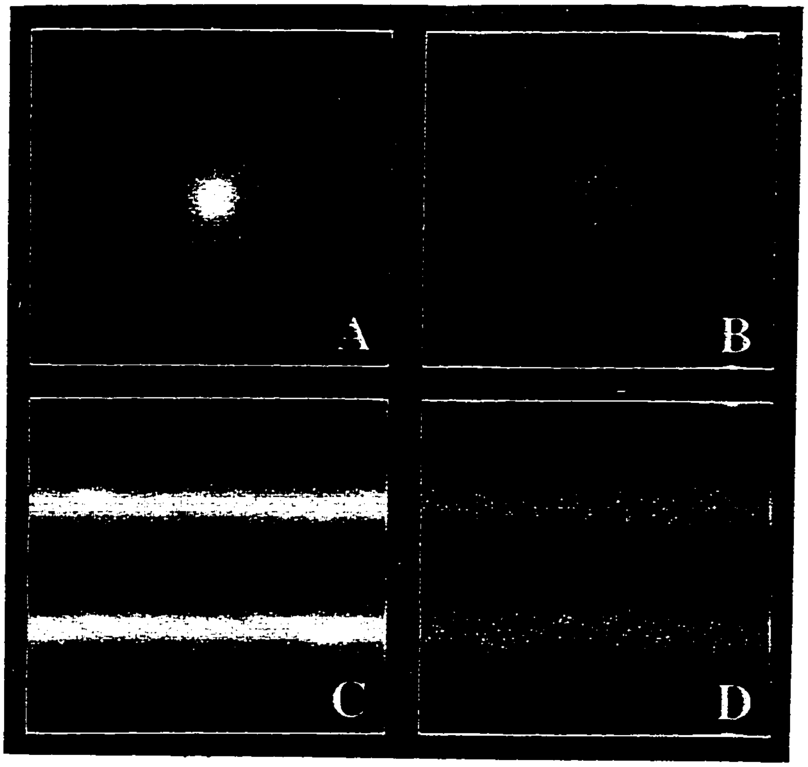


Figure 7.10. The estimated DTFs based on different image detail. (a) The estimated DTF by using the right side of the unoverlapped part for the phantom image with Gaussian low-pass DTFs. (b) The estimated DTF by using the left side of the unoverlapped part for the phantom image with Gaussian low-pass DTFs. (c) The estimated DTF by using the right side of the unoverlapped part for the phantom image with belt Butterworth band-pass DTFs. (d) The estimated DTF by using the left side of the unoverlapped part for the phantom image with belt Butterworth band-pass DTFs.

of the non-rigid motion. In this project, the distances Δ_1 and Δ_2 between the upper and lower ghosts and the central image in the abdomen image were manually measured to be $\Delta_1 = -73$ pixels for the upper ghost and $\Delta_2 = 68$ pixels for the lower ghost. This was done by recognizing some most visible structures or features in both the central image and the ghost and then manually measuring the distance between these two copies of the structures or features. For example the upper boundary of the abdomen were easily seen in both central image and the upper ghost after colormap manipulation. These distances for the upper and lower ghosts from the central image differ unexpectedly by 5 pixels, much greater than any expected experimental error (±1 pixel). The physical mechanism behind this phenomenon is not yet known. However this difference does offer another reason behind the failure of Mitsa's model, which predicts these distances to be identical.

- The estimated DTFs for the abdomen image are shown in Fig. 7.11. The used window positions and window sizes are shown in Fig. 7.12 and were slightly smaller than those used in the phantom observation.

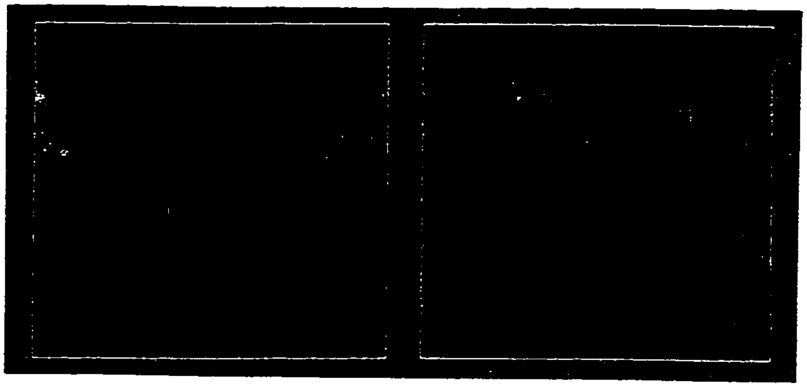


Figure 7.11. The estimated DTFs of upper ghost (left) and lower ghost (right) for the abdomen image

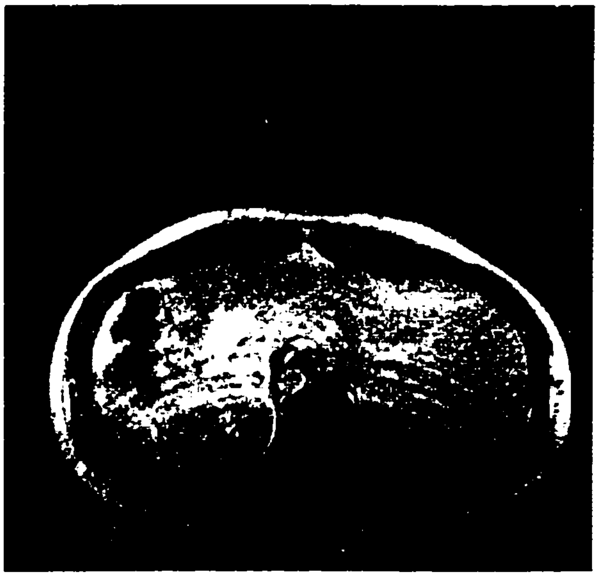


Figure 7.12. The window size and position used for DTF estimation for the abdomen image

The estimated abdomen DTFs in Fig. 7.11 look similar to belt band-pass filters. That agrees with our earlier observation: the ghost outlines of the abdomen image are very much like the ghost outlines displayed in the phantom image with DTF having a form of belt band-pass filter. Again the physical background of this belt band-pass filter characteristics of DTFs of the abdominal motion is not immediately known.

In this project. we also experimented with different assumed positions for the upper (Δ_1) and lower (Δ_2) ghosts. Fig. 7.13(a)(b) and (c) show the estimated DTFs when $\Delta_1 = -73$. $\Delta_2 = 73$; $\Delta_1 = -68$, $\Delta_2 = 68$ and $\Delta_1 = -70$, $\Delta_2 = 70$ respectively. Compared to the estimated DTFs with actually measured distances $\Delta_1 = -73$ and $\Delta_2 = 68$ as shown in Fig. 7.11, there is no significant differences between the estimated DTFs with the different pairs of Δ_1 and Δ_2 in the experiments. They all have similar belt band-pass filter forms.

7.6 Summary

The distortion transfer functions (DTFs) can be estimated from the corrupted image directly by deconvoluting the unoverlapped part of the ghost by the corresponding unoverlapped part of the central image. However for the deconvolution to be stable. windowing techniques are used to avoid the undesirable effects of artificially sharp edges imposed on the unoverlapped parts and to avoid the instabilities induced by noise. The estimated DTFs for phantom images indicate that this method of DTF estimation is efficient provided there is sufficient detail in the image. The estimated DTFs for the abdomen image were found to be belt band-pass filters. This agrees with the observation that the ghost outlines of the abdomen image are very similar to that of the phantom images corrupted by DTFs having forms of belt band-pass filters. The physical explanation to this characteristics of the abdominal motion is not known.

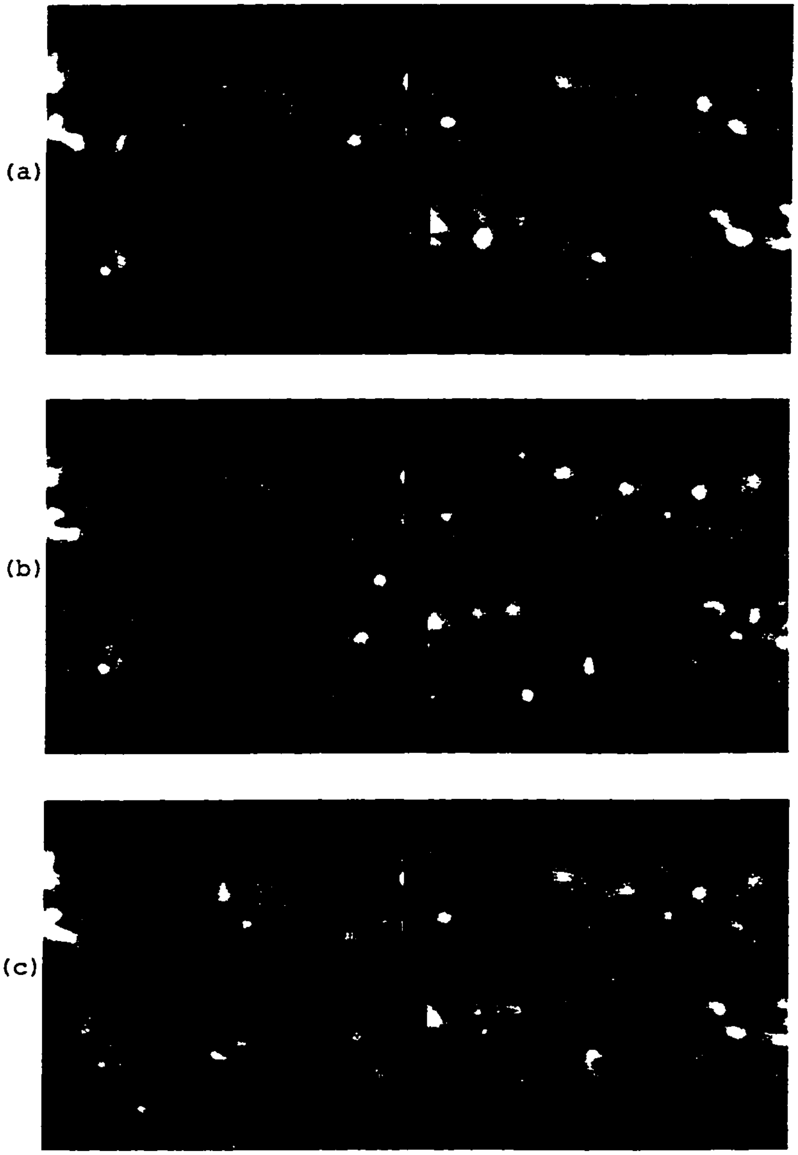


Figure 7.13. The estimated DTFs in the abdomen image with different assumed positions of the upper (Δ_1) and lower (Δ_2) ghosts. The right and left sides are for the upper and lower ghosts respectively. (a) when $\Delta_1 = -73$, $\Delta_2 = 73$; (b) when $\Delta_1 = -68$, $\Delta_2 = 68$; (c) when $\Delta_1 = -70$, $\Delta_2 = 70$.

CHAPTER 8

A COMPOSITE IMAGE PROCESSING APPROACH OF MOTION ARTIFACT SUPPRESSION

8.1 Introduction

A composite image processing method for motion artifact suppression in MRI is proposed in this chapter. Then experiments are designed to examine how the tasks involved in this composite method impact upon the artifact suppression results. Finally the correction results for the real abdomen image are presented.

The submitted paper "Using a MRI Distortion Transfer Function to Characterize Motion of a Non-rigid Object" by Yang and Smith [1998] used some of the results presented in this chapter.

8.2 The Composite Image Processing Method of Motion Artifact Suppression

Motion artifact suppression becomes very easy if the central image (true image) does not overlap with any ghost. In this simple, but not practical case, motion artifacts can be easily removed by putting zeros outside the region of support (ROS) of the object. However in practice the central image and ghosts overlap with each other and the difficulty is how to separate this overlapped part. A composite method of motion artifact suppression is proposed for non-rigid periodic motions along the slice selection axis. The strategy of this composite image processing method is to gradually tune the correct partition for the overlapping between the central image and ghosts by making the estimated central image and ghosts satisfy the generalized motion model regarding variant characteristics of motions in both time and space.

Let's state some notations first. $S_{recorded}$ stands for the recorded (or measured) data in k-space, which is corrupted by motion. $I_{recorded}$ is the reconstructed image of $S_{recorded}$, which can be achieved by applying the discrete Fourier transform on $S_{recorded}$.

S_c and I_c are the frequency domain data and space domain data of the central image respectively. S_g and I_g are the frequency domain data and space domain data of the ghosts respectively. R_c represents the region of support (ROS) of the central image. R_g stands for the ROS of ghosts. $R_{g\cap c}$ is the part of R_g that overlaps with the ROS of the central image. $R_{g\not\cap c}$ is the part of R_g that is outside ROS of the central image. Hence the region of support of ghosts is given by: $R_g = R_{g\cap c} + R_{g\not\cap c}$. This implies the ghosts in space domain can be divided into two parts: $I_{g_{out}}$ and $I_{g_{in}}$, where $I_{g_{out}}$ represents the ghosts part outside the ROS of the central image. $I_{g_{in}}$ is the ghosts part overlapped with the central image. In other words, $I_{g_{in}} = I_g \in R_{g\cap c}$, $I_{g_{out}} = I_g \in R_{g\not\cap c}$. $R_{g_{out}}$, the ghosts part outside the ROS of the central image can be obtained straightforwardly by superimposing $R_{g\not\cap c}$ onto the corrupted image $I_{recorded}$. However, getting $I_{g_{in}}$, the overlapped part of ghosts with ROS of the central image, needs more elaboration.

At the beginning of the composite method, N: the number of ghosts displaying in the corrupted image and the distance Δ_n, $n = 1, 2, ..., N$, between each ghost and the central image must be measured from the corrupted image. Then R_c, $R_{g\cap c}$ and $R_{g\not\cap c}$ are obtained by using the snakes and region labeling algorithms as discussed in Chapter 4-6. Then DTFs for all ghosts are estimated as discussed in Chapter 7.

Assume we have a rough estimate of ghosts I_g^i at the i^{th} iteration. The next estimate I_g^{i+1}, which is expected to be a better estimate of ghosts, can be obtained by two different paths:

- In path 1, we make use of the fact that the ghost part outside the ROS of the central image is known and unchanged. The $(i+1)^{th}$ estimate of ghosts is composed of two parts: the unchanged part $I_{g_{out}}$ and the overlapped part $I^{i+1}_{g_{in_{path1}}}$ from I_g^i. That is:

$$I^{i+1}_{g_{in_{path1}}} = I_g^i \in R_{g\cap c} \qquad\qquad I^{i+1}_{g_{path1}} = I_{g_{out}} + I^{i+1}_{g_{in_{path1}}} \tag{8.1}$$

- By path 2 the generalized motion model and the frequency domain information available from the current estimate of the ghosts are used to generate a new estimate.

$$S^i_g = \mathcal{F}[I^i_g] \qquad\qquad \tilde{S}^i_{c_{path2}} = S_{recorded} - S^i_g$$

where $\tilde{S}^i_{c_{path2}}$ is the i^{th} estimated frequency data of the central image. Note that the ROS of $\tilde{S}^i_{c_{path2}}$'s space domain data, $\tilde{I}^i_{c_{path2}}$, is likely no longer R_c but the whole field of view when S^i_g does not provide a correct estimate of ghosts. The i^{th} estimated central image with ROS bounded can be obtained as:

$$I^i_{c_{path2}} = \tilde{I}^i_{c_{path2}} \in R_c$$

where $\tilde{I}^i_{c_{path2}} = \mathcal{F}^{-1}[\tilde{S}^i_{c_{path2}}]$

The frequency domain data $S^i_{c_{path2}}$ of the i^{th} estimated central image with ROS bounded is:

$$S^i_{c_{path2}} = \mathcal{F}[I^i_{c_{path2}}]$$

Then making use of the generalized motion model Equation 3.4, we have:

$$\begin{aligned} S^{i+1}_{g_{path2}} &= S^i_{c_{path2}} \cdot \sum_{n=1}^{N} DTF_n(k_x, k_y) \exp^{-j2\pi\Delta_n k_y} \\ I^{i+1}_{g_{path2}} &= \mathcal{F}^{-1}[S^{i+1}_{g_{path2}}] \end{aligned} \tag{8.2}$$

The $(i+1)^{th}$ estimate of ghosts can be considered as the average of the estimates from the two different paths:

$$I^{i+1}_g = \frac{1}{2}(I^{i+1}_{g_{path1}} + I^{i+1}_{g_{path2}}) \tag{8.3}$$

That means the $(i+1)^{th}$ estimate of ghosts takes into account information about the generalized motion model and the ROSs and should be a better approximation than the i^{th} estimate provided the used generalized motion model and ROSs are correct.

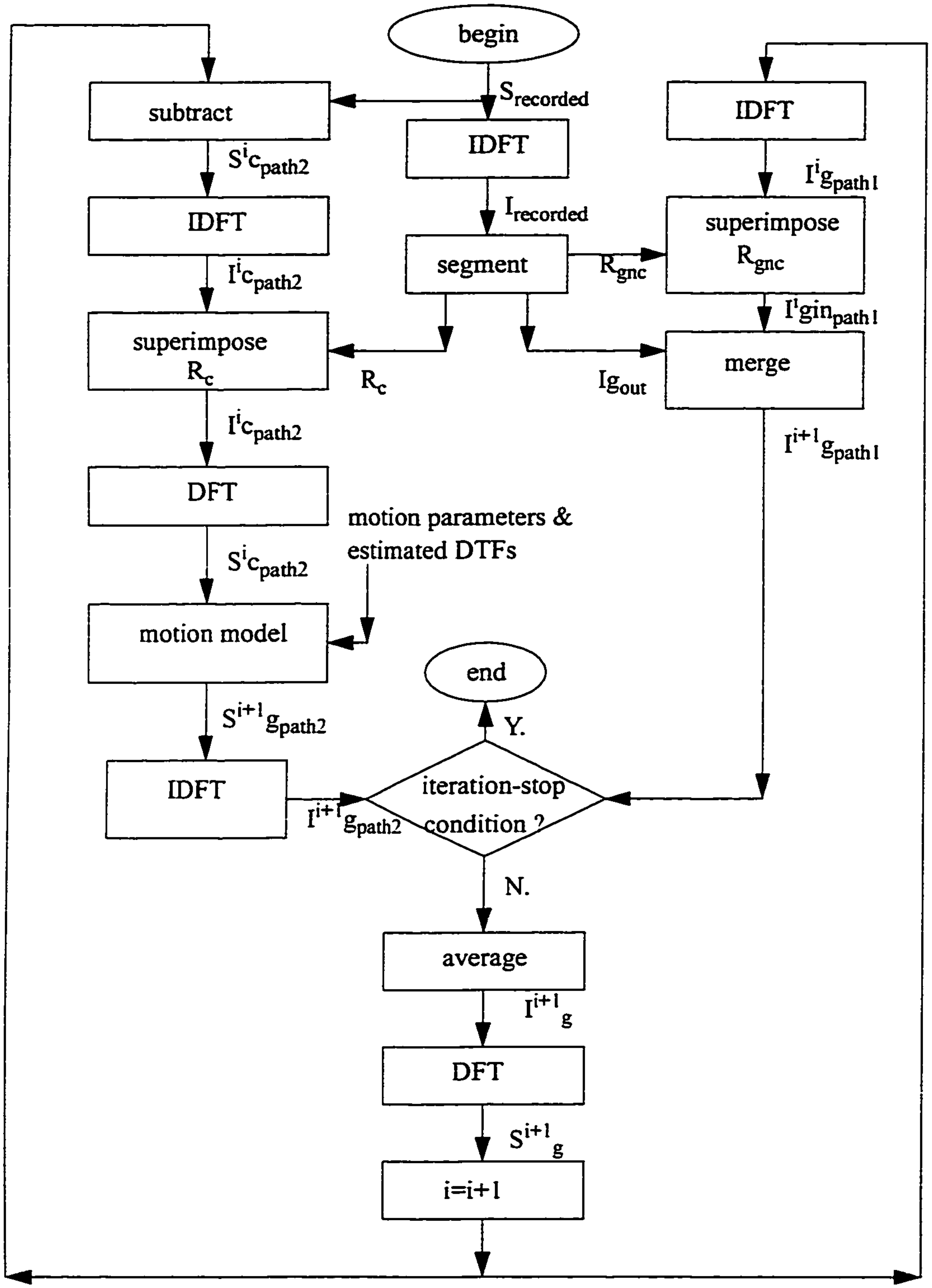


Figure 8.1. Block diagram of the composite image processing method for removing artifacts associated with motion of a non-rigid body

The block diagram of this composite method is shown in Fig. 8.1.

The detail steps of the method are:

1. Inverse Fourier transform the measured motion corrupted k-space data $S_{recorded}$ and obtain the corrupted image $I_{recorded}$.

2. Manually measure N and Δ_n, n=1,2,...,N, for all the ghosts.

3. Apply the snake algorithms and region labeling algorithms to the corrupted image. $I_{recorded}$, to obtain R_c, R_g, $R_{g\cap c}$ and $R_{g\not\cap c}$.

4. Superimpose $R_{g\not\cap c}$ onto the corrupted image, $I_{recorded}$ and obtain $I_{g_{out}}$, which is the portion of ghosts that lies outside the ROS of the central image and considered as accurate.

5. The i^{th} iteration begins with an initial estimate S_g^i, the frequency data for the ghosts. At the beginning of iteration, i = 0, set S_g^0 to zero.

6. By path 1, inverse Fourier transform S_g^i to obtain I_g^i, an estimate of the ghosts in image domain.

7. Superimpose $R_{g\cap c}$ to I_g^i and obtain $I^i_{g_{in_{path1}}}$, a ghost portion that is overlapped with ROS of the central image.

8. Merge $I^i_{g_{in_{path1}}}$ with $I_{g_{out}}$ by Equation 8.1 and get a new estimate of the ghosts: $I^{i+1}_{g_{path1}}$.

9. By path2, subtract S_g^i from the recorded k-space data $S_{recorded}$ and obtain a ROS unbounded central image's frequency data: $\tilde{S}^i_{c_{path2}}$.

10. Inverse Fourier transform $\tilde{S}^i_{c_{path2}}$ to obtain its corresponding image domain data $\tilde{I}^i_{c_{path2}}$. $\tilde{I}^i_{c_{path2}}$ is ROS unbounded, meaning that the ROS of $\tilde{I}^i_{c_{path2}}$ is the whole field of view.

11. Superimpose R_c onto $\tilde{I}^i_{c_{path2}}$ and obtain the ROS bounded image $I^i_{c_{path2}}$.

12. Fourier transform $I^i_{c_{path2}}$ to obtain its frequency data $S^i_{c_{path2}}$.

13. Apply the motion model Equation 8.2 and obtain a new estimate of frequency data for ghosts: $S^{i+1}_{g_{path2}}$.

14. Inverse Fourier transform $S^{i+1}_{g_{path2}}$ to obtain its corresponding space domain data. $I^{i+1}_{g_{path2}}$.

15. Stop the iteration if defined iteration-stop condition is satisfied. In our experimental investigation, since we are not looking for a highly efficient algorithm. we simply chose to stop after a fix number of iterations. $\tilde{I}^i_{c_{path2}}$ is put out as an estimate of the central image for quantitative evaluation. Otherwise average the estimates as Equation 8.3 to obtain I^{i+1}_g, a new estimate of ghosts for next iteration.

16. Fourier transform I^{i+1}_g to obtain its corresponding frequency data S^{i+1}_g. Go back to step 5 and start the $(i+1)^{th}$ iteration.

8.3 Effects of the Tasks Involved in the Composite Method on the Success of Motion Artifact Suppression

As we discussed above, in our composite method for motion artifact suppression there are several image processing tasks involved, including:

1. The estimation of N: number of ghosts and Δ_n, $n = 1, 2, ..., N$, the distance between each ghost and the central image.

2. The estimation of contours of the central image and ghosts followed by segmentation of R_c, $R_{g\cap c}$ and $R_{g\not\cap c}$.

3. The estimation of the distortion transfer function DTF_n; $n = 1, 2, ..., N$, for each ghost.

Generally the success of motion artifact suppression by this composite method is dependent on every task involved. We defined two metrics to evaluate the success of correction in this project.

- The central energy error remaining, $C_{error-remain}$. was defined as the ratio of central difference energy after correction and before correction. $C_{error-remain}$ can be calculated by:

  $$C_{error-remain} =$$

  $$100\% * \sum_{(x,y)\in R_c} |I_{corrected}(x,y) - I_{true}(x,y)|^2 / \sum_{(x,y)\in R_c} |I_{corrupted}(x,y) - I_{true}(x,y)|^2$$

  where R_c is the region of support (ROS) of the central image. $I_{true}(x.y)$. $I_{corrupted}(x.y)$. $I_{corrected}(x,y)$ are the intensities of the true image without any motion artifacts. the motion corrupted image and the corrected image respectively.

  This measure, $C_{error-remain}$, is better than comparing the normalized mean square difference of the corrected and uncorrected images relative to the true image. The normalized mean square difference is grossly biased by the signal energy associated with the portion of the central image undistorted by ghosts. $C_{error-remain}$ shows how much percentage of the remaining difference energy that is still there after correction compared to before correction. If there is no correction, $C_{error-remain}$ stays at 100%. If the correction is perfect, $C_{error-remain}$ becomes zero. This means the distortion in the central image is completely removed so that central part of the corrected image is exactly the same as that of the true image.

  $C_{error-remain}$ is important because it is associated with the distortion in the central image used for diagnosis. Unfortunately $C_{error-remain}$ is not available for real MRI images as true images typically are not available in real MRI applications. To overcome this problem another metric, the ghost energy remaining, was defined.

- The ghost energy remaining, $G_{error-remain}$, was defined as ratio of the ghost energy outside R_c, the region of support of the central image. after correction to before correction. Mathematically $G_{error-remain}$ is given by:

  $$G_{error-remain} = 100\% * \sum_{(x,y)\notin R_c} |I_{corrected}(x,y)|^2 / \sum_{(x,y)\notin R_c} |I_{corrupted}(x,y)|^2$$

  $R_{g-remain}$ is a good indication for the success of ghost suppression outside R_c showing how much percentage of the ghost energy that is still remained after correction compared to before correction. The smaller $R_{g-remain}$ is, the more the artifacts are suppressed. $R_{g-remain}$ becomes zero when the artifact suppression is complete outside the region of support of the central image. While $R_{g-remain}$ stays at 100% when no artifacts are suppressed at all.

  $C_{error-remain}$ is available for both phantom images and real MRI images. If we can show that $C_{error-remain}$ tracks $G_{error-remain}$ for phantom images. we can assume a similar behavior for the medical images. Thus an improvement in $G_{error-remain}$ indicates an expected improvement in $C_{error-remain}$. In many algorithms used to suppress ghosts, $G_{error-remain}$ is artificially set to zero and this relationship between $G_{error-remain}$ and $C_{error-remain}$ is not available. However since our algorithm attempts to characterize both the ghosts and central images, $G_{error-remain}$ is not forced to zero and remain a valid metric of the success of the artifact suppression.

The following experiments are designed to investigate how the tasks impact upon the suppression results.

8.3.1 The iterative approach

For the phantom images corrupted by a variety of distortion transfer functions (DTFs) as shown in Fig. 3.1 - Fig. 3.8 in chapter 3, the motion parameters are

$N = 2$, $\Delta_1 = -80$, $\Delta_2 = 80$. With these parameters and the theoretical DTFs. the corresponding corrected images by using our composite method are shown in Fig. 8.2 - Fig. 8.5. The central energy error remaining $C_{error-remain}$ and the ghost energy remaining $G_{error-remain}$ are listed in Table 8.1.

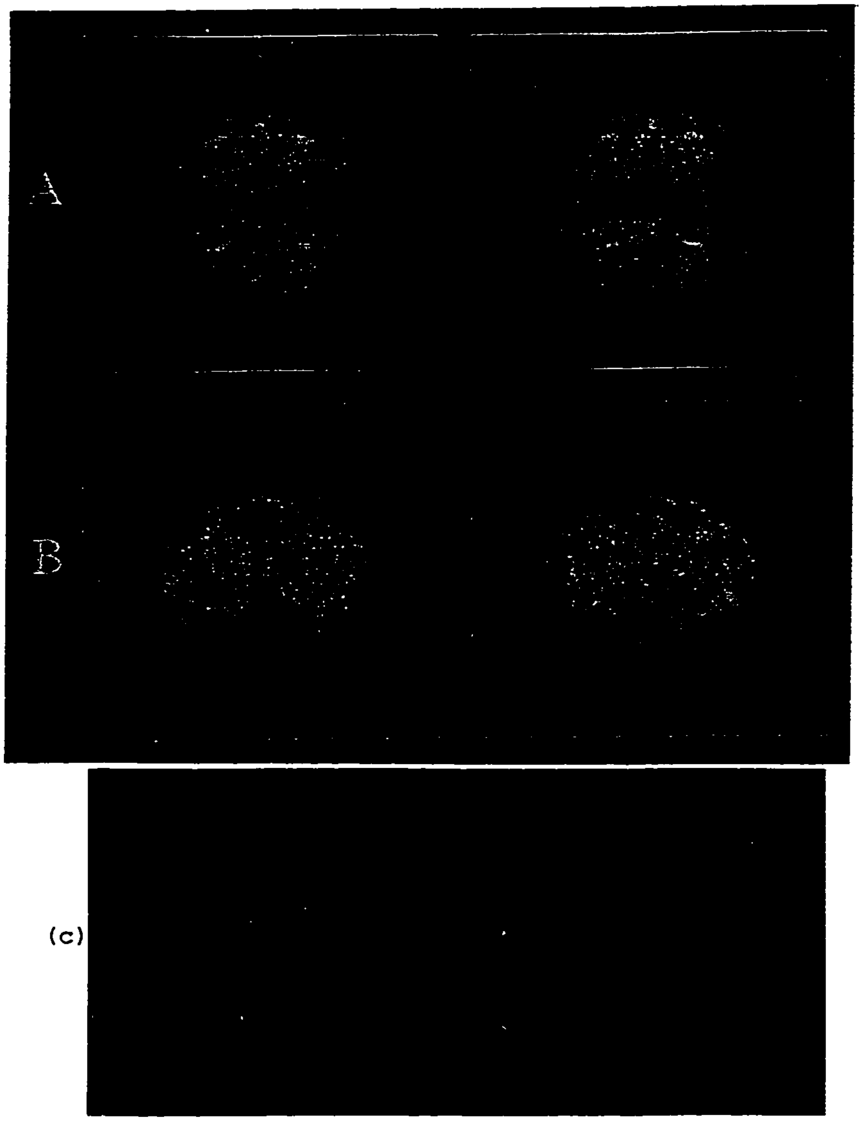


Figure 8.2. (b) shows the corrected images for the phantom images (a) with Gaussian low-pass DTFs (left) and Butterworth low-pass DTFs (right) respectively when the motion parameters and theoretical DTFs are exactly known. (c) shows the difference images between the corrected and the true images (with the intensity scaled up by 65 times)

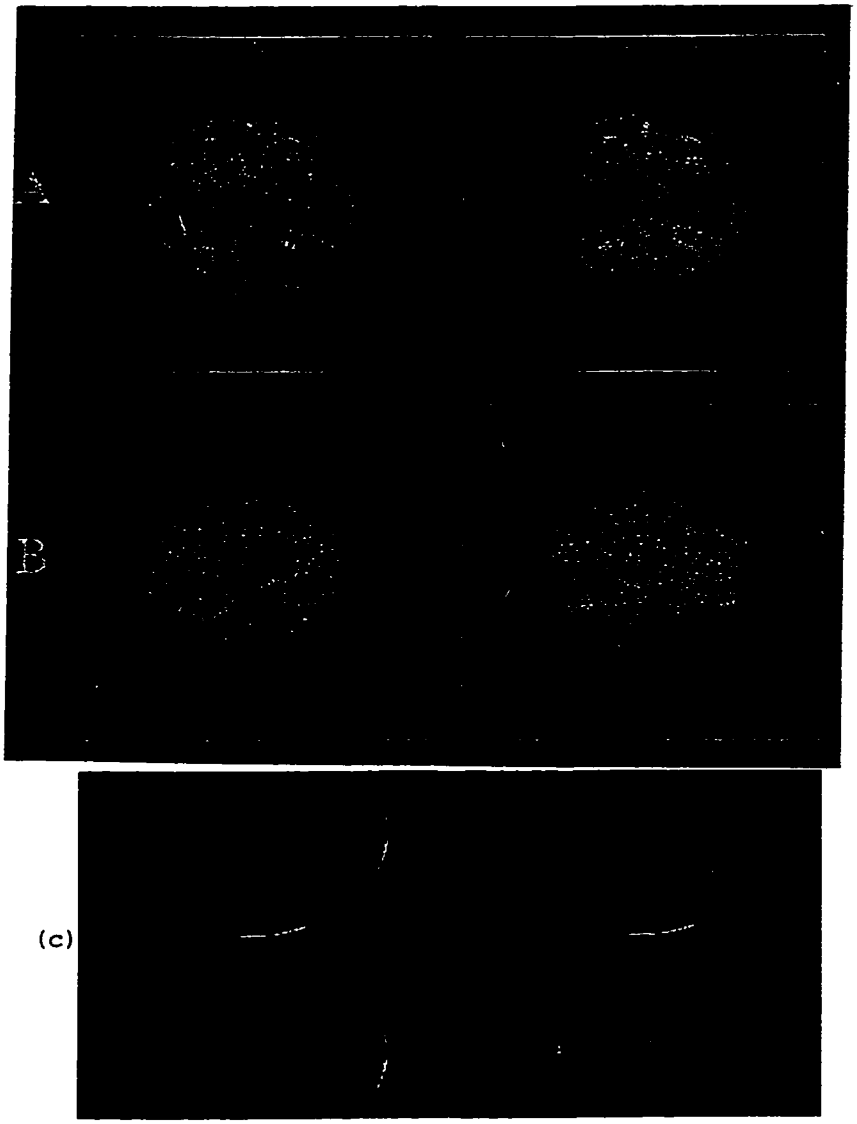


Figure 8.3. (b) shows the corrected images for the phantom images (a) with radially symmetric (left) and belt (right) Butterworth band-reject DTFs respectively when the motion parameters and theoretical DTFs are exactly known. (c) shows the difference images between the corrected and the true images (with the intensity scaled up by 65 times)

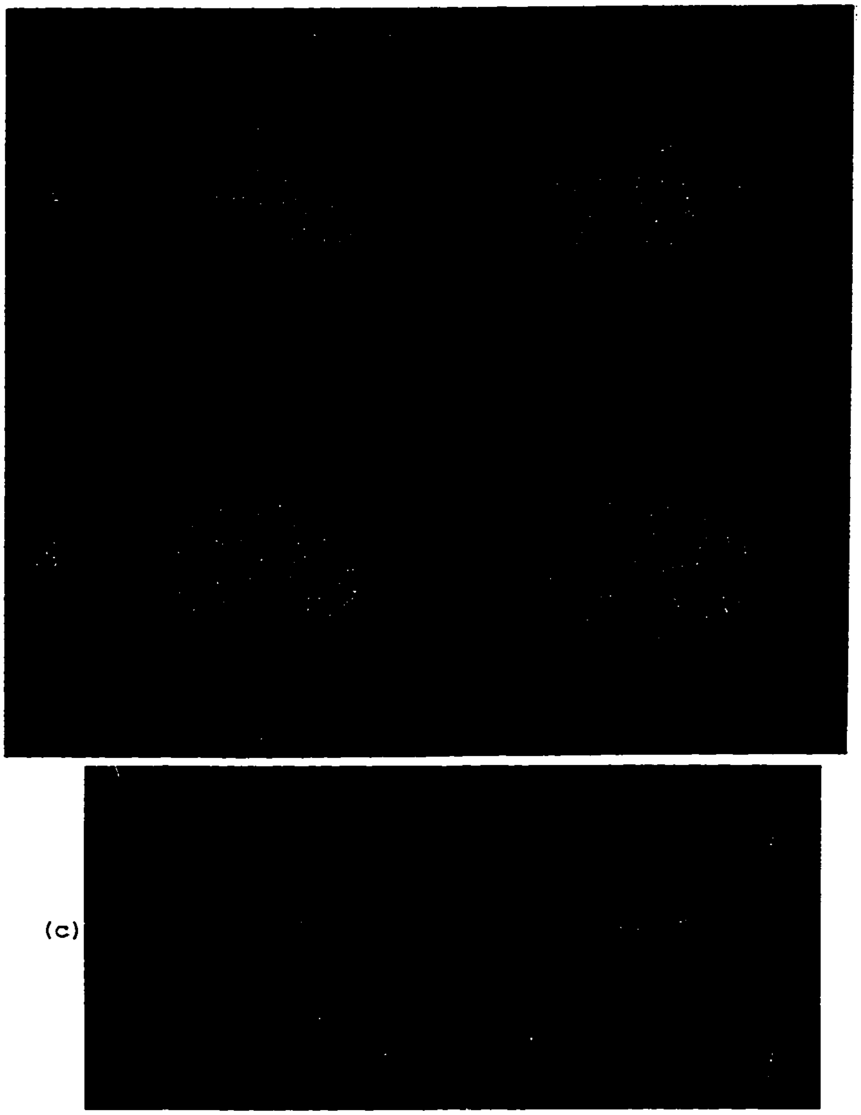


Figure 8.4. (b) shows the corrected images for the phantom images (a) with Gaussian high-pass DTFs (left) and Butterworth high-pass DTFs (right) respectively when the motion parameters and theoretical DTFs are exactly known. (c) shows the difference images between the corrected and the true images (with the intensity scaled up by 65 times)

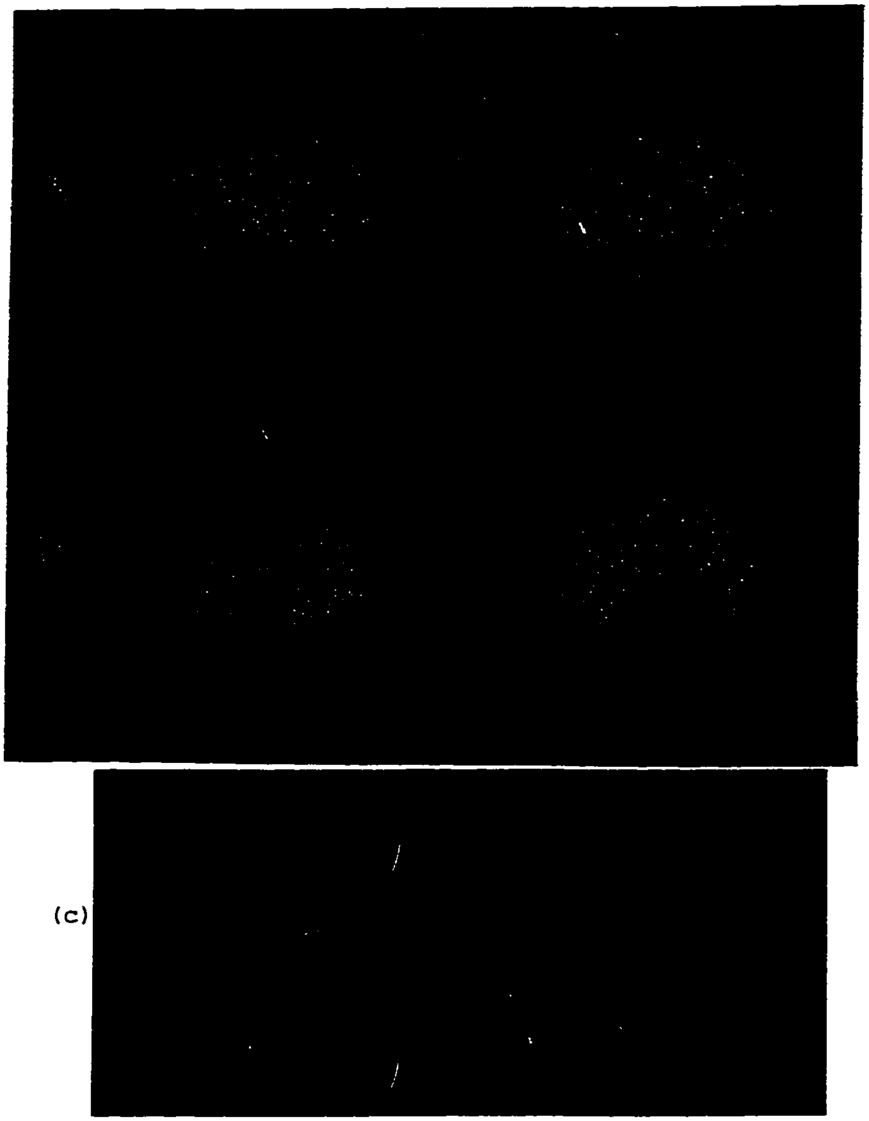


Figure 8.5. (b) shows the corrected images for the phantom images (a) with radially symmetric (left) and belt (right) Butterworth band-pass DTFs respectively when the motion parameters and theoretical DTFs are exactly known. (c) shows the difference images between the corrected and the true images (with the intensity scaled up by 65 times)

DTFs' form	$C_{error-remain}$(%)	$G_{error-remain}$(%)
Gaussian low-pass filter	0.0009	0.003
Butterworth low-pass filter	0.002	0.004
Butterworth band-reject filter (radially symmetric)	0.007	0.006
Butterworth band-reject filter (belt)	0.007	0.004
Gaussian high-pass filter	0.4	0.7
Butterworth high-pass filter	0.5	0.7
Butterworth band-pass filter (radially symmetric)	0.6	1.3
Butterworth band-pass filter (belt)	0.5	1.5

Table 8.1. The central energy error remaining $C_{error-remain}$ and ghost energy remaining $G_{error-remain}$ of the corrections when DTFs and parameters about the motions are exactly known

As shown in Fig. 8.2 - Fig. 8.5, the distortion associated with the non-rigid motions are reduced. The small objects that were blurred by motions and hardly seen in the corrupted images become clearer. That suggests the iterative approach is successful. Detailed examination shows that the correction results for the phantom images corrupted by DTFs having a form of low-pass and band-reject filters are better than those having a form of high-pass and band-pass filters. The metrics $C_{error-remain}$ and $G_{error-remain}$ in Table 8.1 agree with this observation. We think this limitation is mainly due to the merging of the latest estimate $I_{g_{in}}^{i}$ of the ghost image that had been overlapped by the central image, with the known, accurate ghost image data $I_{g_{out}}$. Simple replacement of the data $I_{g_{in}}^{i}$ into the original ghost image works well. but has a tendency to generate a central image with the outer ghost contours superimposed upon it. This is because the two parts of the ghost image are discontinuous. which generates artifacts in k-space that disrupt the iteration's convergence. However a smoothing merging is difficult to achieve across the changing contour between the overlapped and non-overlapped portions of the images.

Note that the ghost energy remaining $G_{error-remain}$ tracks the central error remaining $C_{error-remain}$ well as shown in Table 8.1. An improvement in $C_{error-remain}$ always goes with an improvement in $G_{error-remain}$. $C_{error-remain}$ was from 0.3 to 2.0 times of $G_{error-remain}$. In the experiments, we also defined C_{MSE}, the normalized mean square difference of corrected image relative to the true image as:

$$C_{MSE} = 100.0\% * \sum_{(x,y)\in R_c} |I_{corrected}(x,y) - I_{true}(x,y)|^2 / \sum_{(x,y)\in R_c} |I_{true}|^2$$

we found that this normalized mean square difference was a poor indication for the success of ghost suppression. For example, our visual system and the difference images between the corrected images and the true image all indicated that improvement for the phantom image with Gaussian low-pass DTFs was much better than that with belt Butterworth band-pass DTFs. However the C_{MSE} numbers, 0.0005% for the phantom

DTFs' form	$C_{error-remain}$(%)	$G_{error-remain}$(%)
Gaussian low-pass filter	0.009	0.004
Butterworth low-pass filter	0.02	0.006
Butterworth band-reject filter (radially symmetric)	0.03	0.01
Butterworth band-reject filter (belt)	0.02	0.006
Gaussian high-pass filter	1.0	0.9
Butterworth high-pass filter	1.3	0.9
Butterworth band-pass filter (radially symmetric)	3.7	3.0
Butterworth band-pass filter (belt)	3.5	2.3

Table 8.2. The central energy error remaining $C_{error-remain}$ and ghost energy remaining $G_{error-remain}$ of the corrections when the estimated DTFs are used

image with Butterworth belt band-pass DTFs and 0.0002% for the phantom image with Gaussian low-pass DTFs, both were very small. The reason was these C_{MSE} numbers did not take into account the severity degree of corruption before correction. Actually the normalized mean square difference of uncorrected image relative to the true image was 22.3% and 0.09% for the phantom images with Gaussian low-pass DTFs and belt Butterworth band-pass DTFs respectively. That is the corruption for the phantom image with Gaussian low-pass DTFs was more severe than that for the phantom image with belt band-pass DTFs.

8.3.2 Effects of estimation of DTF

Fig. 8.6 and Fig. 8.7 show the corrected images under the same conditions as in section 8.3.1 except the used DTFs are not the theoretical ones but the estimated ones as shown in Fig. 7.2 - Fig. 7.9 in Chapter 7. The corresponding central energy error remaining $C_{error-remain}$ and ghost energy remaining $G_{error-remain}$ are listed in Table 8.2.

These results indicate that the more accurate the DTFs used during the artifact suppression, the better the correction results. We have noted that great detail in the unoverlapped parts of the image leads to better estimate of the DTFs. In addi-

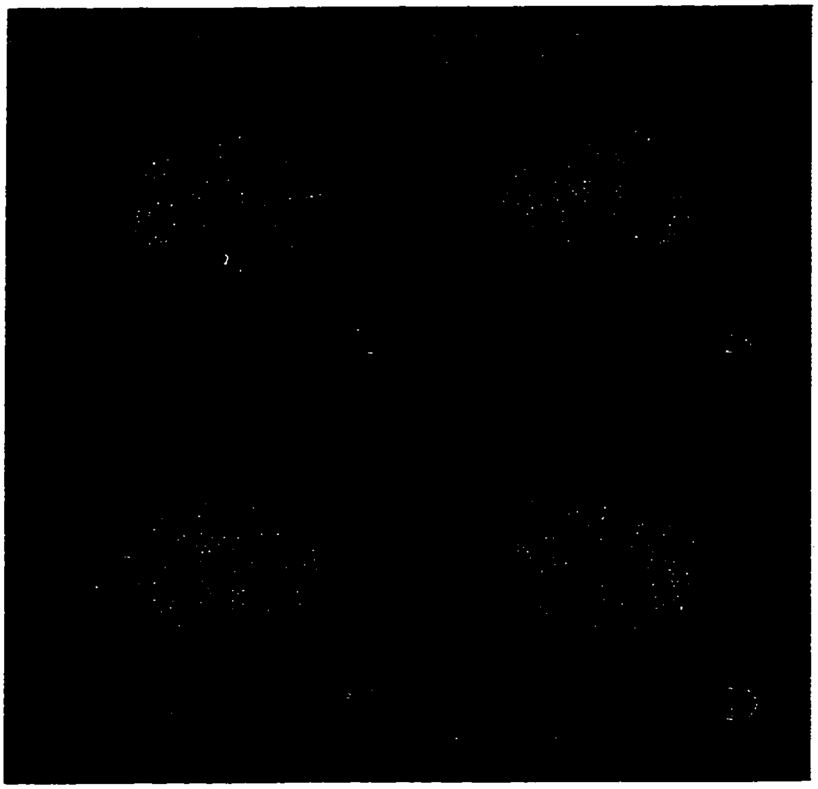


Figure 8.6. The corrected images for the phantom images with DTFs having a form of Gaussian low-pass filter (a), Butterworth low-pass filter (b), radially symmetric Butterworth band-reject filter (c) and belt Butterworth band-reject filter (d) respectively when the used DTFs are not theoretical but estimated

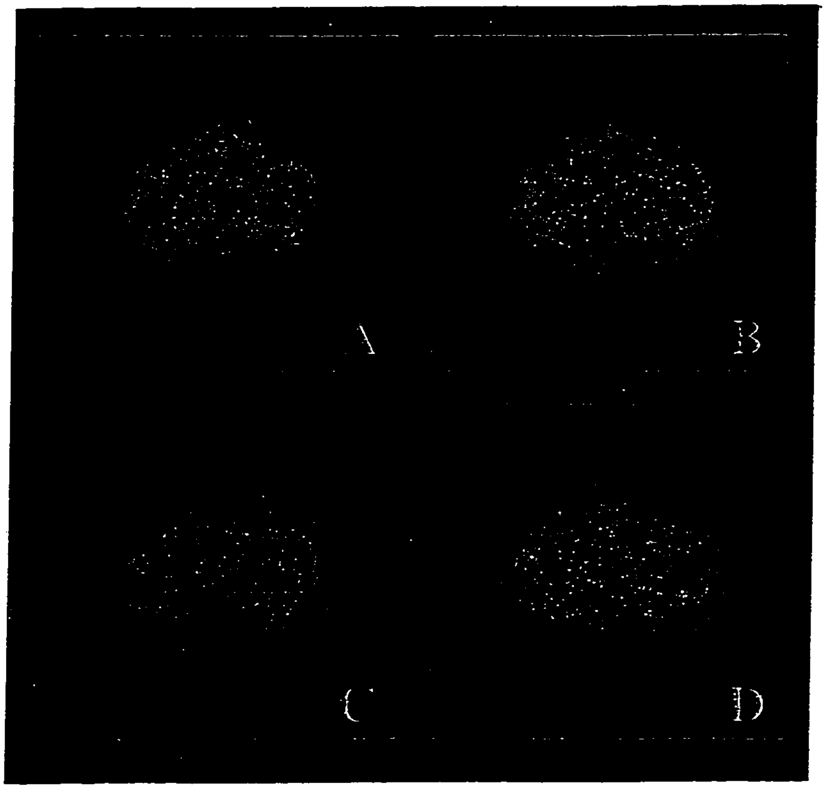


Figure 8.7. The corrected images for the phantom images with DTFs having a form of Gaussian high-pass filter (a), Butterworth high-pass filter (b), radially symmetric Butterworth band-pass filter (c) and belt Butterworth band-pass filter (d) respectively when the used DTFs are not theoretical but estimated

tion. we can also see from Table 8.2 that an improvement in ghost energy remaining $G_{error-remain}$ goes with an improvement in central energy error $C_{error-remain}$.

8.3.3 Effects of contours

As discussed earlier in Chapter 4 - 5, the snakes are gradually attracted to the desired contour from far away even at places where edges are very weak. However not all the points are exactly at the sharpest edges along the contour because of the global optimization of the snakes. We investigated the effects of inaccurracy in the contour upon motion artifact correction. Fig. 8.8 and Fig. 8.9 show the correction results for the phantom images with DTFs having a form of Gaussian low-pass filter as shown in Fig. 3.1 and belt Butterworth band-pass filter as shown in Fig. 3.8 respectively. The same conditions as in section 8.3.1 are used except that the contours of the central images were deliberately made several pixels smaller or bigger. The corresponding central energy error remaining $C_{error-remain}$ and ghost energy remaining $G_{error-remain}$ are listed in Table 8.3.

The results indicate that the corrections become even better when the snaked contour of the central image is slightly inflated 1-3 pixels in terms of $C_{error-remain}$ and $G_{error-remain}$. On the other hand the corrected image appearance, $C_{error-remain}$ and $G_{error-remain}$ all indicate that the corrections become much worse when the contours are deflated. That is because the distortion in the corrected image is more obvious when the region of support (ROS) goes towards inside the object than that when ROS goes towards outside the object. Therefore in this project we inflated the snaked contour about 3 pixels for the composite correction algorithm to ensure that none of the points along the contour goes inside the object. However we kept the ghost contours in order to avoid causing extra inflation. The contour inflation and deflation were done by using the contour ballooning algorithm discussed earlier in Chapter 6. Again it is noticed the greatly decreased sensitivity of the approach when trying to

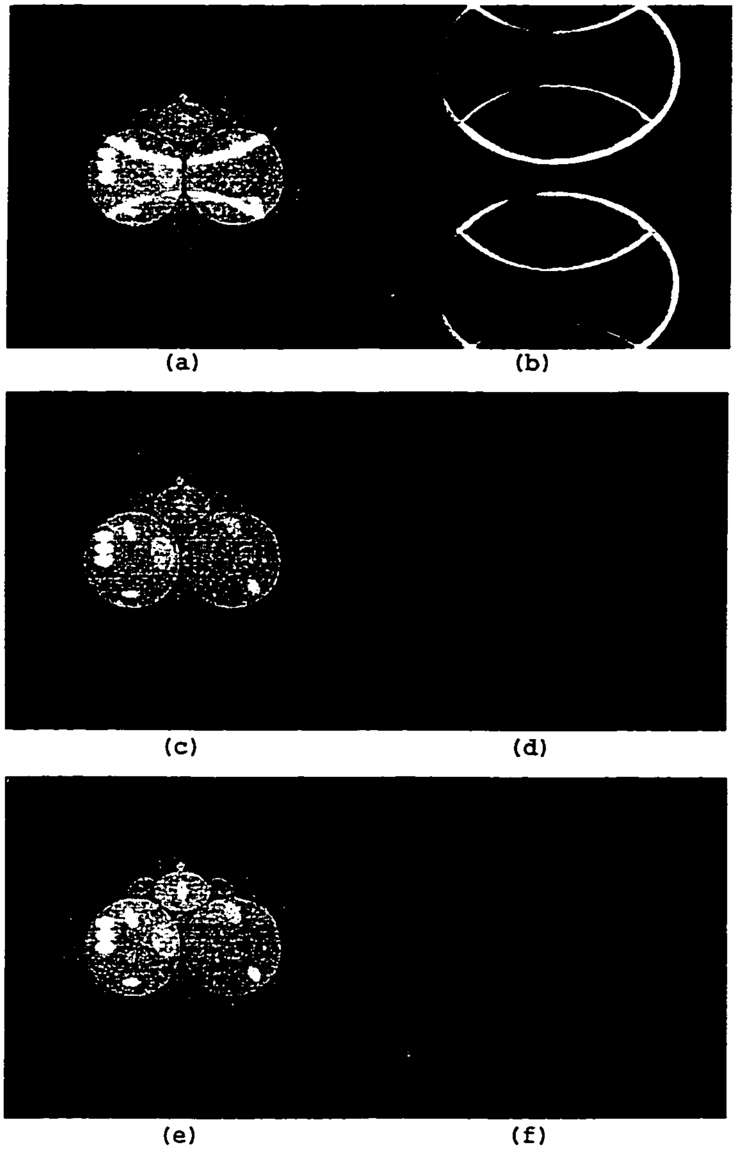


Figure 8.8. The corrected (left) and their difference (right, intensity scaled up by 10 times) images between the corrected and the true images for the phantom image with Gaussian low-pass DTFs when the contours are deliberately made several pixels smaller or bigger. (a)(b) Using the snake contour deflated by 3 pixels. (c)(d) Using the snake contour. (e)(f) Using the snake contour inflated by 3 pixels.

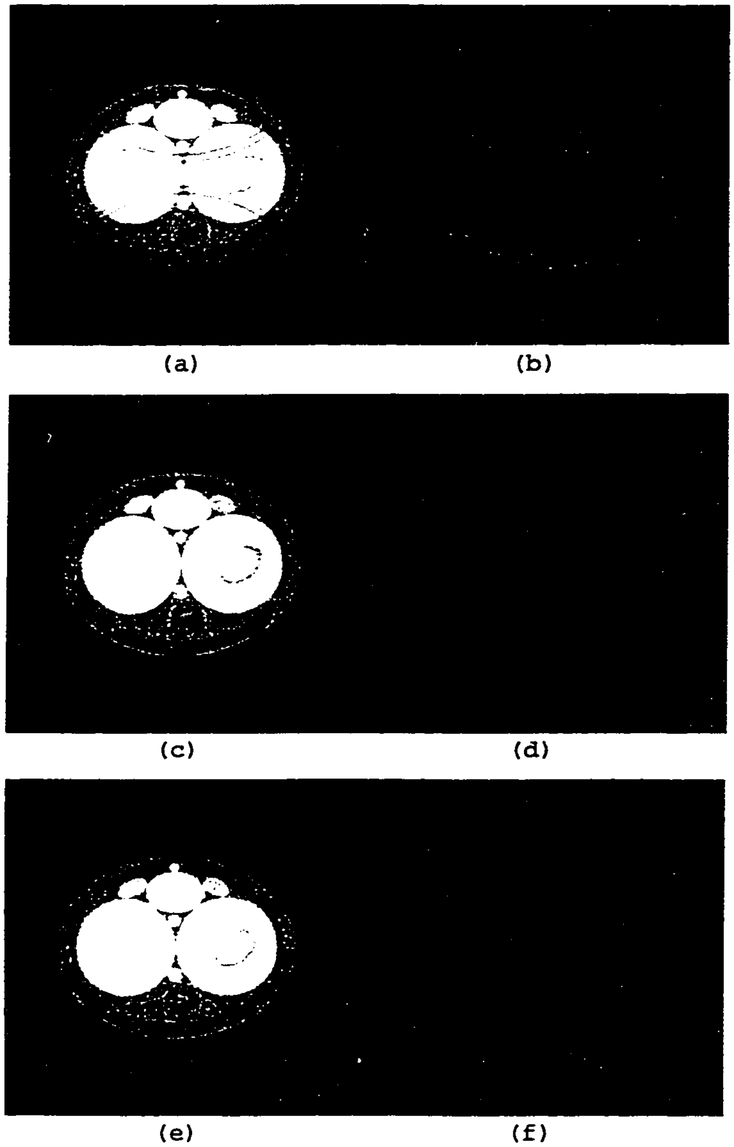


Figure 8.9. The corrected (left) and their difference (right, intensity scaled up by 10 times) images between the corrected and the true images for the phantom image with belt Butterworth band-pass DTFs when the contours are deliberately made several pixels smaller or bigger. (a)(b) Using the snake contour deflated by 3 pixels. (c)(d) Using the snake contour. (e)(f) Using the snake contour inflated by 3 pixels.

Form of DTF	Δ (pixels)	$C_{error-remain}$ (%)	$G_{error-remain}$ (%)
Gaussian low-pass filter	3	0.0009	0.003
Gaussian low-pass filter	2	0.0005	0.01
Gaussian low-pass filter	1	0.003	0.2
Gaussian low-pass filter	0	0.03	0.2
Gaussian low-pass filter	-1	0.2	1.1
Gaussian low-pass filter	-2	0.9	2.2
Gaussian low-pass filter	-3	2.1	3.5
Butterworth band-pass filter (belt)	3	0.5	1.5
Butterworth band-pass filter (belt)	2	1.0	4.7
Butterworth band-pass filter (belt)	1	2.6	41.6
Butterworth band-pass filter (belt)	0	10.4	49.1
Butterworth band-pass filter (belt)	-1	60.5	63.0
Butterworth band-pass filter (belt)	-2	69.1	58.5
Butterworth band-pass filter (belt)	-3	256.6	57.2

Table 8.3. The central energy error remaining $C_{error-remain}$ and ghost energy remaining $G_{error-remain}$ of the corrections when the snaked contour are inflated ($\Delta > 0$) and deflated ($\Delta < 0$) Δ pixels

Form of DTF	Distance (pixels)	$C_{error-remain}$ (%)	$G_{error-remain}$ (%)
Gaussian low-pass filter	77	1.0	0.1
Gaussian low-pass filter	78	0.5	0.06
Gaussian low-pass filter	79	0.07	0.01
Gaussian low-pass filter	80	0.009	0.004
Gaussian low-pass filter	81	0.07	0.009
Gaussian low-pass filter	82	0.3	0.03
Gaussian low-pass filter	83	0.6	0.05
belt Butterworth band-pass filter	77	7.1	3.2
belt Butterworth band-pass filter	78	5.5	2.9
belt Butterworth band-pass filter	79	4.7	2.7
belt Butterworth band-pass filter	80	3.5	2.3
belt Butterworth band-pass filter	81	4.4	2.4
belt Butterworth band-pass filter	82	5.6	2.5
belt Butterworth band-pass filter	83	7.8	2.8

Table 8.4. The central energy error remaining $C_{error-remain}$ and ghost energy remaining $G_{error-remain}$ of the corrections when the distance between the upper ghost and the central image is not accurate

correct phantom images with DTFs of simple characteristics.

8.3.4 Effects of inaccurately determining distances between ghosts and the central image

If the distances, Δ_n, $(n = 1, 2, ..., N)$, between ghosts and the central image are not accurate, how are the estimated DTFs and the iterative process affected? Experiments were made under the same conditions as in Section 8.3.2 except that the distance Δ_n used for estimating DTF and for the iterative algorithm were deliberately made one or two pixels smaller or bigger. Fig. 8.10 and Fig. 8.11 show the corrected and their difference images compared to the true image for the phantoms with DTFs having a form of Gaussian low-pass filter and belt Butterworth band-pass filter respectively. The corresponding central energy error remaining $C_{error-remain}$ and ghost energy remaining $G_{error-remain}$ are listed in Table 8.4.

The results indicate the following:

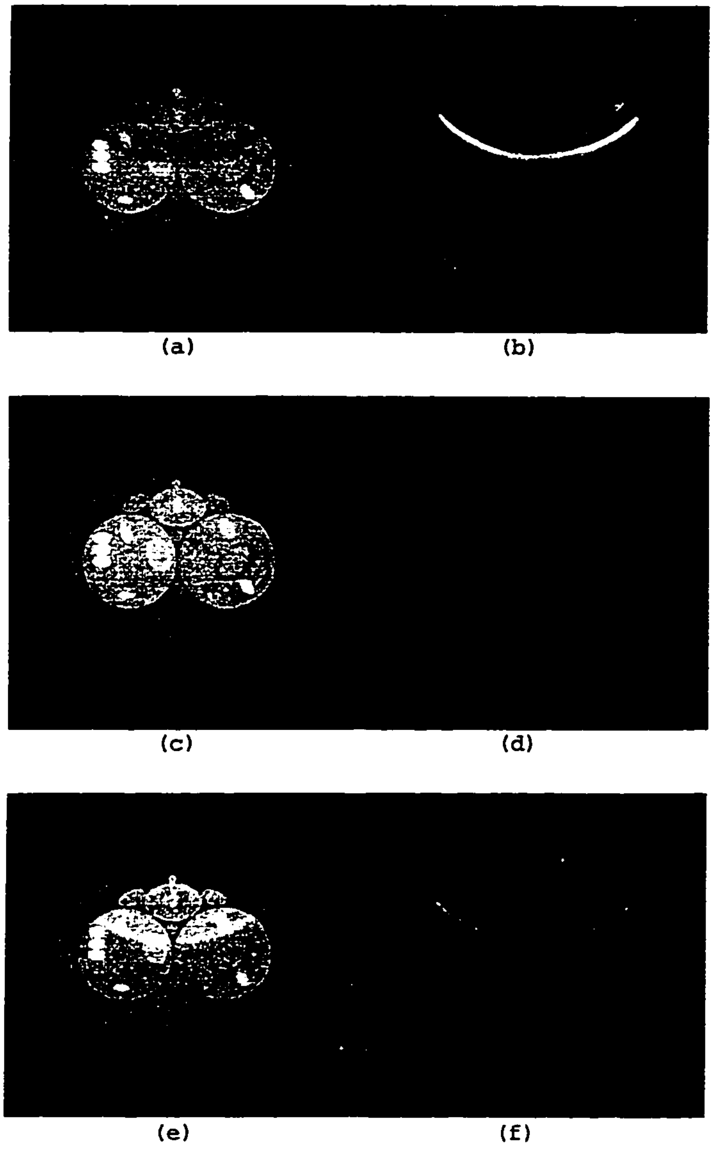


Figure 8.10. The corrected (left) and the difference (right, intensity scaled up by 10 times) images compared to the true image for the phantom image with Gaussian low-pass DTFs when the distance between the upper ghost and the central image is not accurate. (a)(b) The used distance = 77. (c)(d) The used distance = 80 (accurate). (e)(f) The used distance = 83.

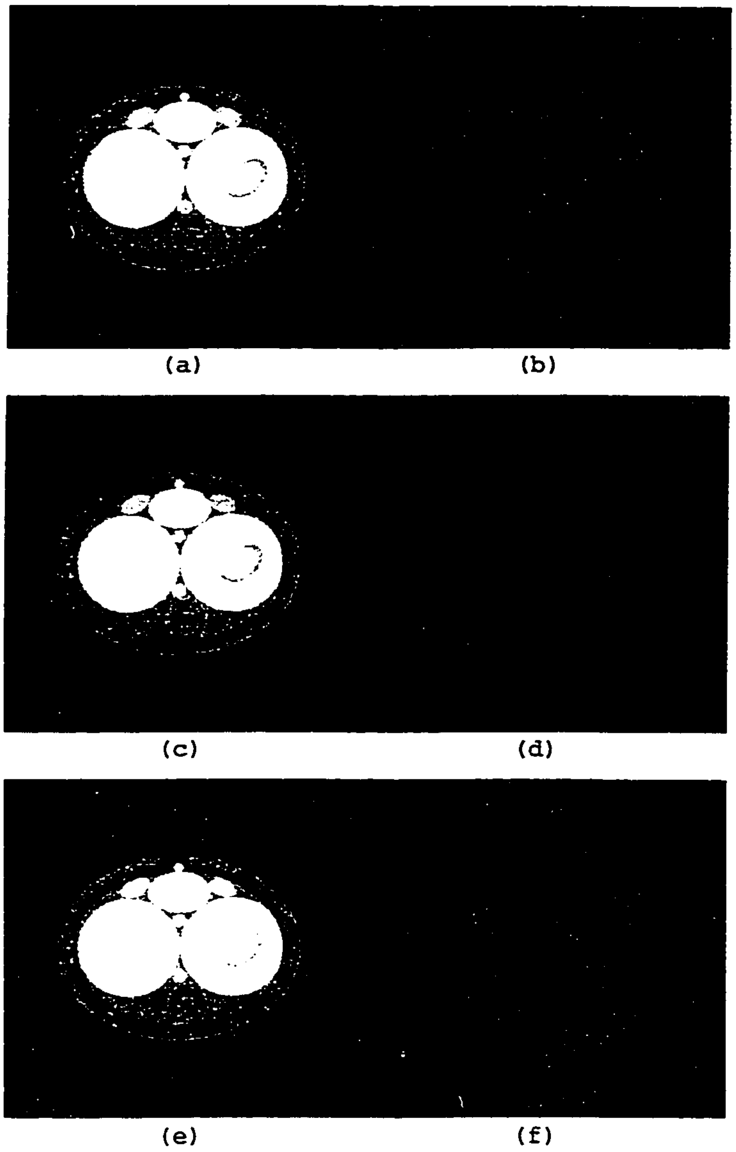


Figure 8.11. The corrected (left) and the difference (right, intensity scaled up by 10 times) images compared to the true image for the phantom image with belt band-pass DTFs when the distance between the upper ghost and the central image is not accurate. (a)(b) The used distance = 77. (c)(d) The used distance = 80 (accurate). (e)(f) The used distance = 83.

- The composite method still worked even when the estimation of the distances was not accurate. As shown in Table 8.4, the greatest central energy error remaining is 7.8% and the greatest ghost energy remaining is 3.2%.

- The correction with inaccurate distances is not as successful as that with accurate distances. That is error in estimates of the relationships between ghosts and the central image can not lead to perfect artifact suppression by the composite method.

8.4 The results with the Real Abdomen Image

To demonstrate the feasibility of the generalized motion model and the composite image processing method, the abdomen image in Fig. 2.5 was tested.

For comparison, the correction results by using Mitsa's method are shown in Fig. 8.12. The central energy error remaining is not available as the true image is not known. The ghost energy remaining was calculated to be 99.2%, meaning ghost energy is hardly suppressed outside the region of support of the central image. It can be seen from the corrected image that the central image almost stayed the same, meaning there is little correction.

As discussed earlier, from the corrupted image, two major ghosts can be seen clearly. The distances between the ghosts and the central image in the abdomen image are manually measured to be 73 pixels for the upper ghost and 68 pixels for the lower ghost. The DTFs estimated directly from the abdomen image were shown in Fig. 7.11 in Chapter 7. The snaked contour for the central image and the contours of ghosts obtained by inflating that of the central image are shown in Fig. 5.4 and Fig. 5.6 respectively in Chapter 5. The region of support of the central image, R_c, is obtained by our region labeling approach and was shown in Fig. 6.3 in Chapter 6. By using our composite method with these motion parameters, DTFs for the upper and lower ghosts and region information about the central image and ghosts, the correction results were obtained as shown in Fig. 8.13. The ghost energy remaining is 31.7% as shown in Table 8.5, meaning less than 1/3 ghost energy remained and about 2/3 of the ghost energy outside the region of support of the central image was removed. From the corrected image as Fig. 8.13(b), it is obvious that the ghost energy is significantly suppressed.

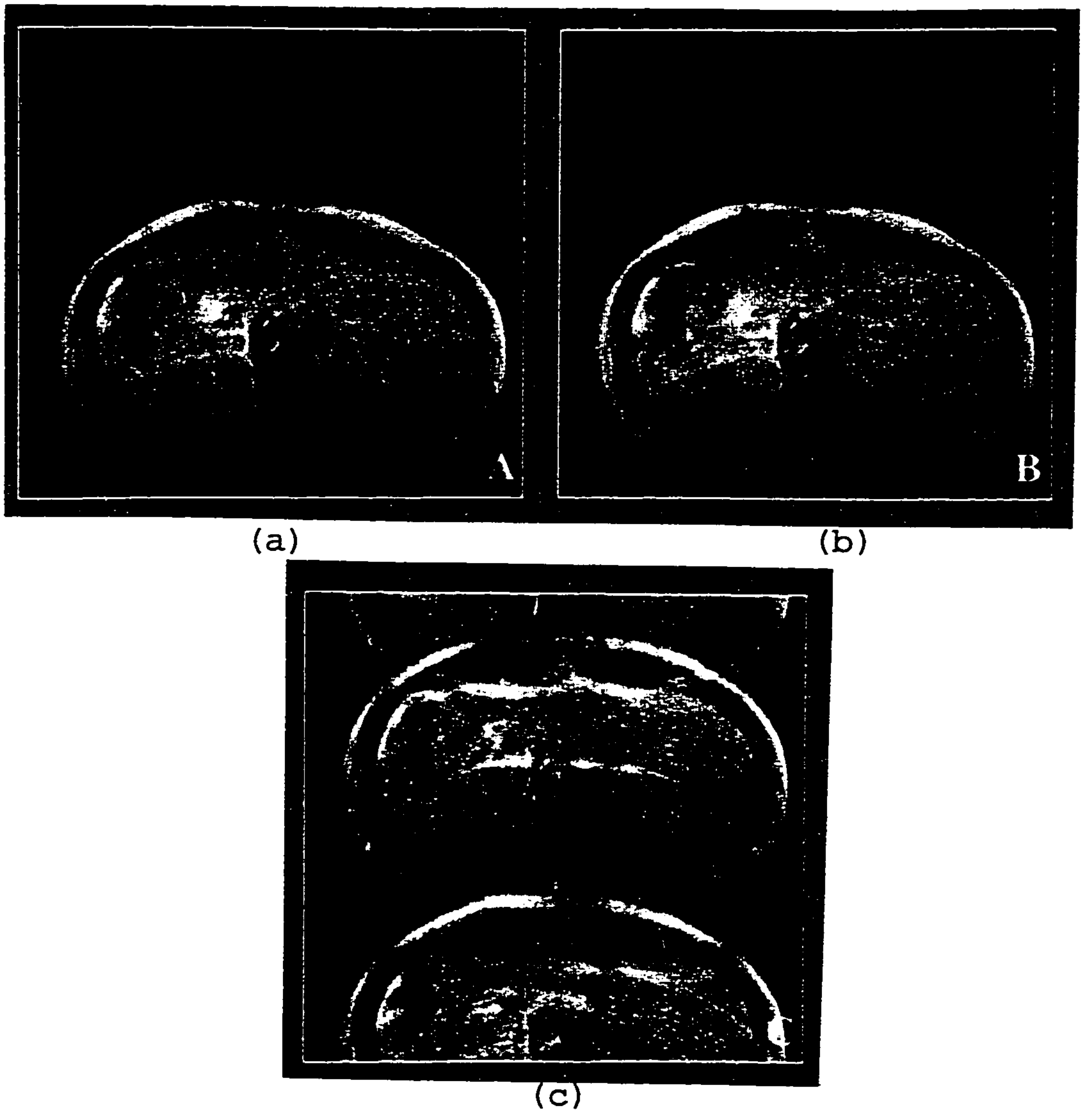


Figure 8.12. The corrected image for the abdomen image by using Mitsa's method. (a) The corrupted abdomen image. (b) The corrected image by using Mitsa's method. (c) the difference image between (a) and (b) with intensity scaled up by 70 times. This small difference image indicate little change in image

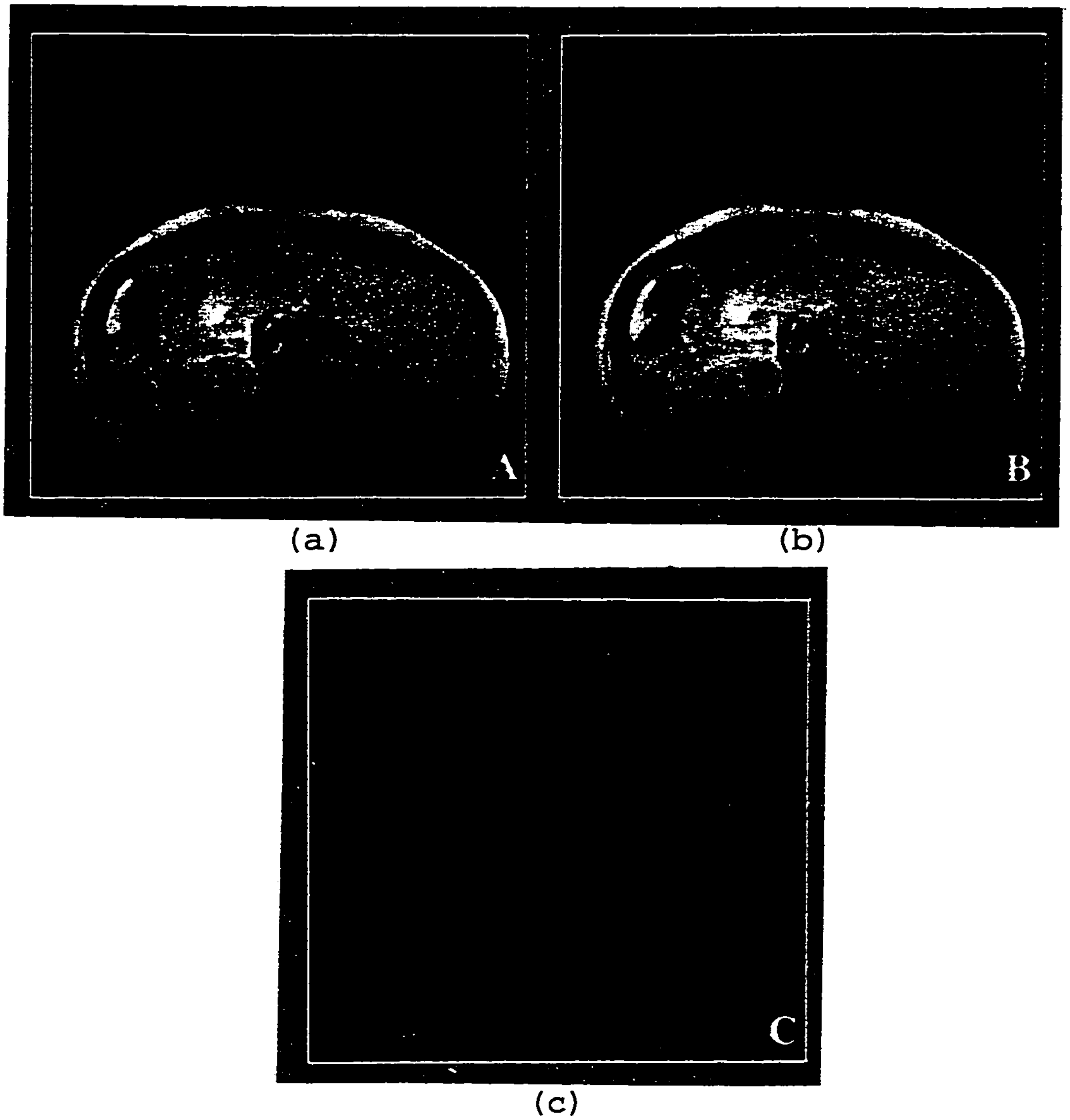


Figure 8.13. The corrected image for the abdomen image by using our composite method with $\Delta_1 = -73$ and $\Delta_2 = 68$. (a) The corrupted abdomen image. (b) The corrected image by using our composite method. (c) the difference image between (a) and (b) with intensity scaled up by 7 times

Motion artifact suppression method	$C_{error-remain}$ (%)	$G_{error-remain}$ (%)
Mitsa's method		99.2
Our composite method		31.7

Table 8.5. The ghost energy remaining $G_{error-remain}$ of the corrections by Mitsa's method and our composite method

With the same conditions as for Fig. 8.13 except different distances Δ_1 and Δ_2 between the ghosts and the central image and different estimated DTFs, we experimented the composite method of motion artifact suppression for the abdomen image. We used $\Delta_1 = -73$, $\Delta_2 = 73$ and the corresponding estimated DTFs shown in Fig. 7.13(a) for Experiment 1, $\Delta_1 = -68$ and $\Delta_2 = 68$ and the corresponding estimated DTFs shown in Fig. 7.13(b) for Experiment 2, $\Delta_1 = -70$, $\Delta_2 = 70$ and the corresponding estimated DTFs shown in Fig. 7.13(c) for Experiment 3 respectively. The corrected and difference images are shown in Fig. 8.14(a)(b) and (c). The ghost energy remainings were measured to be 30.3%, 32.5% and 31.5% for Experiment 1, 2, 3 respectively. These results basically agree with our observations to the phantom images discussed in Section 8.3.4 except the correction for the abdomen image is less successful than that for the phantom images. The author think there are at least three possible factors related to the reason why the correction for this abdomen image is less successful than that for the phantom images:

- The accuracy of the estimated DTFs. The unoverlapped parts of the abdomen central image and ghosts don't seem to have enough detail to provide information of the motion in all directions.

- The way of merging in the iteration algorithm. As we pointed out when we discussed the phantom image with DTFs having a form of belt Butterworth band-pass filter, there is a tendency to disrupt the iteration's convergence when simple replacement was done for merging.

- The complex characteristics of the motion. It may be possible there exist more than one non-rigid motion across the abdomen. In this project, only one non-rigid motion was taken into account.

Further research work is certainly required to investigate all possible factors involved in the complicated motion artifact suppression problem for real MRI images.

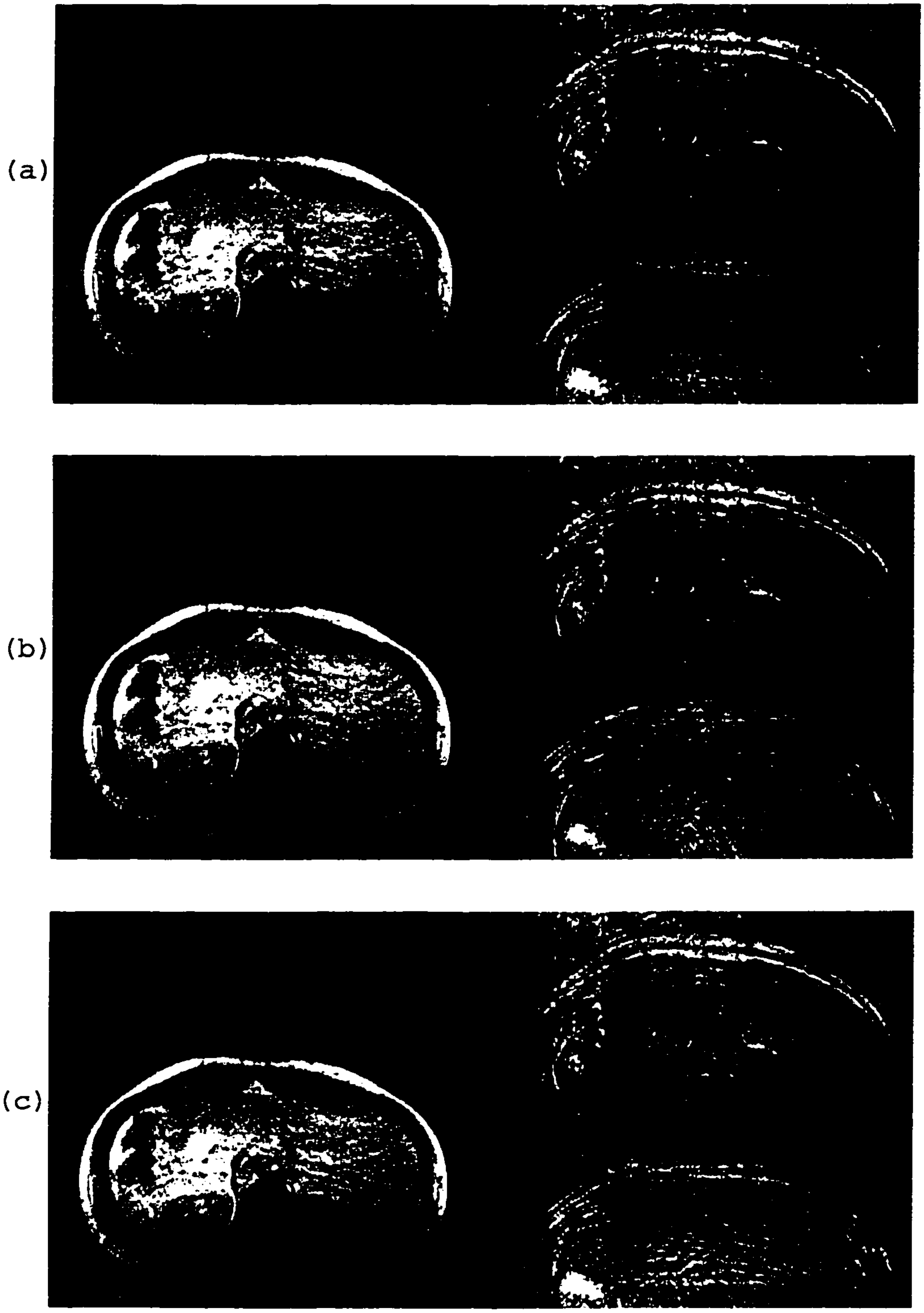


Figure 8.14. The corrected abdomen images (left) and the difference images (right, with intensity scaled up by 7 times) between the corresponding corrected images and the original abdomen image by using our composite method with different assumed positions for the upper (Δ_1) and lower (Δ_2) ghosts. (a) when $\Delta_1 = -73$, $\Delta_2 = 73$; (b) when $\Delta_1 = -68$, $\Delta_2 = 68$; (c) when $\Delta_1 = -70$, $\Delta_2 = 70$.

8.5 Summary

The composite image processing method of motion artifact suppression described in this Chapter is based on the generalized motion model, regions of support information obtained by the snake algorithms and estimates of the DTFs. The success of motion artifact suppression is therefore related to all these image processing tasks. For fast evaluating the success of motion artifact suppression, we defined the central energy error remaining $C_{error-remain}$ and the ghost energy remaining $G_{error-remain}$. This composite method gave excellent results for severely motion corrupted phantom images when the estimates were accurate: The central energy error remaining $C_{error-remain}$ was 0.0009-0.007% and ghost energy remaining was 0.003-0.006% for phantom images with DTFs having forms of low-pass and band-reject filters; For phantom images with DTFs having forms of high-pass and band-pass filters, $C_{error-remain}$ and $G_{error-remain}$ were 0.4-0.6% and 0.7-1.5% respectively. Under the same condition except that motions' parameters and DTFs were estimated rather than theoretical, the correction results were not as perfect as in the case of theoretical DTFs but still very good: $C_{error-remain}$ and $G_{error-remain}$ were 0.009-0.03% and 0.004-0.01% respectively for the phantom images with DTFs having forms of low-pass and band-reject filters; For phantom images with DTFs having forms of high-pass and band-pass filters, $C_{error-remain}$ and $G_{error-remain}$ were 1.0-3.7% and 0.9-3.0% respectively. As to the real abdomen images with unknown respiratory motion, the motion artifacts were significantly suppressed by our composite method with the ghost energy remained of 31.7%, compared to that of 99.2% by Mitsa's method. Although further research work is required for better correction, these results indicated that the generalized motion model and the composite image processing method had a promising potential for correcting both phantom images and the abdomen image with unknown respiratory motion.

CHAPTER 9

CONCLUSION AND FUTURE WORK

In this thesis, post-processing methods for motion artifact suppression in magnetic resonance imaging (MRI) were investigated. When existing post-processing methods were reviewed, it was clear that many existing methods modeled human body motions as rigid despite the obvious fact that the human body typically was not rigid. These oversimplified, thus nonrealistic, motion models did not lead to adequate corrections when applied to real MR images.

Based on experimental observations, a new generalized motion model was empirically proposed. The non-rigidity or spatially variant characteristics of motions was taken into account by introducing a distortion transfer function (DTF). The existing motion models were shown as just special cases of this generalized model. A composite image processing method was proposed to correct ghosts caused by non-rigid periodic motions along the slice selection axis. There were several image processing tasks involved in the composite method, including contour detection and contour-based region labeling and estimation of DTFs for ghosts.

As the outer boundaries of motion corrupted MR images were often weak, even broken at some places, contours detected by conventional contour detection methods, such as mask operators, were not continuous. To solve the contour continuity problem, a new technique, the snake, was investigated. The strategy of snakes was to locate positions of a minimum energy field. The energy field was actually an integration of different kinds of energy along the entire contour. There were usually three kinds of energy: image energy from image features, external energy from extraneous knowledge or constraints and internal energy from the smoothness constraints about the contour. The advantages of snakes included: flexible mechanism to incorporate all kinds of information about the boundary; contour connectivity; and robustness to noise. For computation, discretized snake models were required. The solution of a discretized snake model was obtained by solving a linear equation system evolutionarily. In the implementation of snakes, some issues were considered, such as contour representation, contour resampling, forces choosing, elasticity and rigidity coefficients optimization, image gradient smoothing and stabilities. Continuous contours were achieved for the abdomen image by our implementation of the snakes.

With contours identified, a method, called a generalized scan-conversion method, was proposed by the author to label regions. This generalized scan-conversion method was based on the odd-parity rule from computer graphics. The problem began with a few sampled points on a contour. For the success of the generalized scan-conversion method, a complete set of points on a contour was required, which was achieved by using the cubic B-spline curve fitting and interpolating algorithms. This generalized scan-conversion region labeling method was fast.

A method of estimating distortion transfer functions (DTF) directly from the corrupted images was proposed. The DTFs were estimated as the deconvolution of the unoverlapped part of the ghost and the corresponding unoverlapped part of the central image. To avoid the undesirable effects of artificially sharp edges imposed on the unoverlapped parts and to reduce the instabilities induced by noise, windowing techniques were employed. Phantom images corrupted by a variety of mathematical DTFs were used. The estimated DTFs of the phantom images indicated this method of DTF estimation was effective. The estimated DTFs for the abdomen image looked like belt band-pass filters, which agreed with the observation that the ghost outlines of the abdomen image were very similar to that of the phantom images corrupted by DTFs of belt band-pass filters. The physical explanation to this characteristics of the abdominal motion is not known.

The strategy of the composite image processing method for ghost suppression was to make use of the estimated DTFs and the regions of support (ROS) of the central image and ghosts. With this knowledge it was possible to make the estimated central image and the estimated ghosts satisfy the generalized motion model. Pictures of the corrected images and some associated metrics were provided. We defined central energy error remaining as ratio of the difference energy in the region of support of the central image after correction to before correction. The ghost energy remaining was defined as ratio of ghost energy outside the region of support of the central image after correction to before correction.

This composite method gave excellent results for severely motion corrupted phantom images when the estimates were accurate: The central energy error remaining $C_{error-remain}$ was 0.0009-0.007% and ghost energy remaining was 0.003-0.006% for phantom images with DTFs having forms of low-pass and band-reject filters; For phantom images with DTFs having forms of high-pass and band-pass filters. $C_{error-remain}$ and $G_{error-remain}$ were 0.4-0.6% and 0.7-1.5% respectively. Under the same condition except that motions' parameters and DTFs were estimated rather than theoretical. the correction results were not as perfect as in the case of theoretical DTFs but still very good: $C_{error-remain}$ and $G_{error-remain}$ were 0.009-0.03% and 0.004-0.01% respectively for the phantom images with DTFs having forms of low-pass and band-reject filters; For phantom images with DTFs having forms of high-pass and band-pass filters. $C_{error-remain}$ and $G_{error-remain}$ were 1.0-3.7% and 0.9-3.0% respectively. As to the real abdomen images with unknown respiratory motion, the motion artifacts were significantly suppressed by our composite method with the ghost energy remained of 31.7%, compared to that of 99.2% by Mitsa's method. Although further research work is required for better correction, these results indicated that the generalized motion model and the composite image processing method have a promising potential for correcting both phantom images and the abdomen image with unknown respiratory motion.

In short, the contributions of this project were:

1. Generalizing the existing motion models by introducing a distortion transfer function (DTF).

2. Proposing methods to accurately estimate DTFs for non-rigid motions directly from corrupted images.

3. Investigating and implementing the active contour models: snakes and implementing these snake algorithms for contour detection for corrupted MR images for motion artifact suppression purpose.

4. Proposing a new approach for motion corrupted MRI image segmentation by using the snake contour information.

5. Proposing a new generalized scan-conversion algorithm to label regions given a few points on a contour.

6. Proposing a composite image processing method to remove ghosts caused by non-rigid period motions along the slice selection axis.

The framework of this composite image processing method were demonstrated feasible for both phantom images and the abdomen image with unknown respiratory motions. However, a number of limitations in the implementation of this composite method were identified, which remain to be investigated in future work in this area.

1. How to automatically estimate the number, position and intensity of ghosts in the corrupted images.

2. The implied deconvolution associated with DTF estimation requires applications of windows [Harris 1978] to maintain stability. The problem of automating the

determination of the optimal window shape or characteristics remains to be investigated.

3. The accuracy of estimation of distortion transfer function (DTF) depends on the details present in the non-overlapped portion of the ghost and central image. An approach to identifying the optimal size and location of these areas is required.

4. Theoretical guidelines to determine the elasticity coefficient and rigidity coefficient needed for the snake algorithm should be investigated further to get more stable and more reproducible contours.

5. By the generalized scan-conversion region labeling method, more considerations should be made in case of more complex contours in order to avoid any exceptions to the odd-parity rule.

6. Although we can accurately estimate the DTFs having a form of high-pass and band-pass filters, the reason behind our ability to correct the corresponding images with less success than for images with DTFs having a form of low-pass and band-reject filters needs to be investigated.

A more interesting challenge would be to extend this composite post-processing framework of motion artifact suppression.

1. It would be worthwhile to investigate how to estimate motions' parameters, such as number of major and minor ghosts and distances between the ghosts and the central image, and how to estimate motions' distortion transfer functions when motion's periodicity becomes less obvious when the ghosts overlap more severe and when the motion artifacts become more blurred.

2. It would be an attractive topic to investigate the possibilities of using the estimated distortion transfer functions (DTFs) for valuable medical diagnosis pur-

pose.

3. It would be extremely valuable if different non-rigid motions across the imaged object are taken into account. In our current work, all parts of the imaged object are assumed to undertake an identical non-rigid motion. However in practical cases, it is possible that different parts of the imaged object may undertake different non-rigid motions. Then mathematically the distortion transfer functions should the function of position. That is in the generalized motion model in Equation 3.2, $F_{non-rigid}(k_x, k_y)$ should be $F_{non-rigid}(k_x, k_y, x, y)$. What is the best approach to incorporate this information into the corrected image? This however, is likely to be an extremely difficult task.

BIBLIOGRAPHY

Agathoklis, P. and L. T. Bruton (1982). Practical-BIBO Stability of N-dimensional Discrete Systems. *IEEE Proceedings, Vol. 130*, 236–242.

Amini, A. A., T. E. Weymouth and R. C. Jain (1990). Using Dynamic Programming for Solving Variational Problems in Vision. *IEEE Trans. on Pattern Analysis and Machine Intellegence, Vol. 12*, 855–867.

Atalar, E. and L. Onural (1992). A Respiratory Motion Artifact Reduction Method in Magnetic Resonance Imaging of the Chest. *IEEE Trans. on Medical Imaging, Vol. 10*, 11–24.

Bruton, L. T. and N. Bartley (1985). Three-Dimensional Image Processing Using Concept of Network Resonance. *IEEE Trans. on Circuits and Systems, Vol. CAS-33*, 664–672.

Bruton, L. T. and N. Bartley (1986). The Enhancement and Tracking of Moving Objects in Digital Images Using Adaptive Three-Dimensional Recursive Filters. *IEEE Trans. on Circuits and Systems, Vol. CAS-33*, 604–612.

Cheney, W. and D. Kincaid (1994). *Numerical Mathematics and Computing.* Brooks/Cole Publishing Company.

Cohen, L. D. (1991). Note on Active Contour Models and Balloons. *CVGIP: Image Understanding, Vol. 53*, 211–218.

Constable, R. T. (1997). Improved Activation Maps Via the Elimination of Motion Effects Through Time-Domain Mixing of Data in Conventional Gradient Echo Functional MRI. *MRM, Vol. 38*, 628–637.

Dierckx, P. (1982). Algorithms for Smoothing Data with Periodic and Parametric Splines. *Computer Graphics and Image Processing, Vol. 20*, 171–184.

Ehman, R. C. and J. P. Felmlee (1989). Adaptive Technique for High-Definition MR Imaging of Moving Structures. *Radiology*, *Vol. 173*, 255–263.

Felmlee, J. P., R. L. Ehman, S. J. Riederer and H. W. Korin (1991). Adaptive Motion Compensation in MR Imaging without Use of Navigator Echoes. *Radiology*, *Vol. 179*, 139–142.

Foley, J. D., A. V. Dam, S. K. Feiner, and J. F. Hughes (1996). *Computer Graphics.* Addison-Wesley Publishing Company.

Hamilton, B. (1982). *Medical Diagnostic Imaging Systems.* R and S Press.

Harris. F. J. (1978). On the Use of Windows for Harmonic Analysis. *Processings of the IEEE*, *Vol. 66*, 51–85.

Hedley. M. and H. Yan (1992). Suppression of Slice Selection Axis Motion Artifacts in MRI. *IEEE Trans. on Medical Imaging*, *Vol. 11*, 233–237.

Hedley, M., H. Yan, and D. Rosenfield (1991a). An Improved Algorithm for 2D Translational Motion Artifact Correction. *IEEE Trans. on Medical Imaging*, *Vol. 10*, 548–553.

Hedley, M., H. Yan, and D. Rosenfield (1991b). Motion Artifact Correction in MRI Using Generalized Projections. *IEEE Trans. on Medical Imaging*, *Vol. 10*, 40–45.

Kak, A. C. and M. Slaney (1988). *Principles of Computerized Tomographic Imaging.* IEEE Press.

Kass, M., A. Witkin and D. Terzopoulos (1988). Snakes: Active Contour Models. *International Journal of Computer Vision*, 321–331.

Korin. H. W., F. Farzaneh, R. C. Wright and S. J. Riederer (1989). Compensation for Effects of Linear Motion in MR Imaging. *Magnetic Resonance in Medicine*, *Vol. 12*, 99–113.

Korin, H. W., J. P. Felmlee, R. L. Ehman and S. J. Riederer (1990). Adaptive Technique for Three-dimensional MR Imaging of Moving Structures. *Radiology*, *Vol. 177*, 217–221.

Kruger, D., G. Slavin, R. Muthupillai, R. Grimm and S. Riederer (1997). An Orthogonal Correlation Algorithm for Ghost Reduction in MRI. *MRM*, *Vol. 38*, 678–686.

Langenberger, K. W. and E. Moser (1997). Nonlinear Motion Artifact Reduction in Event-trigged Gradient-echo FMRI. *Magnetic Resonance Imaging*, *Vol. 15*, 163–167.

Levine, M. D. (1985). *Vision in Man and Machine*. McGraw-Hill Book Company.

Lobregt, S. and M. A. Viergever (1995). A Discrete Dynamic Contour Model. *IEEE Trans. on Medical Imaging*, *Vol. 14*, 12–24.

Macgowan, C. K. and M. L. Wood (1996). Phase-encode Reordering to Minimize Errors Caused by Motion. *MRM*, *Vol. 35*, 391–398.

Mitsa, T.. K. J. Parker, W. E. Smith, A. M. Tekalp and J. Szumowski (1990). Correction of Periodic Motion Artifacts along the Slice Selection Axis in MRI. *IEEE Trans. on Medical Imaging*, *Vol. 9*, 310–317.

Ranganath, S. (1995). Contour Extraction from Cardiac MRI Studies Using Snakes. *IEEE Trans. on Medical Imaging*, *Vol. 14*, 328–338.

Riek, J. K. (1994). *Modeling, Detection and Suppression of Motion Artifacts in Magnetic Resonance Imaging*. Ph.D. Thesis, Univ. of Rochester.

Riek, J. K., A. M. Tekalp, W. E. Smith and E. Kwok (1995). Out-of-plane Motion Compensation in Multislice Spin-echo MRI. *IEEE Trans. on Medical Imaging, Vol. 14*, 464–470.

Schultz, C. L., R. J. Alfidi and A. D. Nelson (1984). The Effect of Motion on Two-dimensional Fourier Transformation Magnetic Resonance Images. *Radiology, Vol. 15*. 117–121.

Shung, K. K., M. B. Smith and B. M. W. Tsui (1992). *Principles of Medical Imaging.* Academic Press, Inc.

Slichter, C. P. (1990). *Principles of Magnetic Resonance* (The Third Edition). Springer-Verlag.

Smith, M., L. Chen, H. Yan, J. Zeng, T. Mathews and J. Yang (1997). Alternatives to the Use of the DFT in MRI and Spectroscopic Reconstruction. *International Journal of Imaging Systems*, *Vol. 8*, 558–564.

Smith, M. R., S. T. Nichols, R. Henkelman and M. Wood (1986). Application of Autoregressive Moving Average Parametric Modeling in Magnetic Resonance Image Reconstruction. *IEEE Trans. on Medical Imaging*, *Vol. 5*, 132–139.

Smith, M. R. and W. F. Yang (1998). Approaches for the Generation of MRI Phantoms for the Evaluation of Protocols. *in progress.*

Staib, L. H. and J. S. Duncan (1992). Boundary Finding with Parametrically Deformable Models. *IEEE Trans. on Image Processing*, *Vol. 14*, 1061–1075.

Wang, M., J. Evans, L. Hassebrook and C. Knapp (1996). A Multistage, Optimal Contour Model. *IEEE Trans. on Image Processing*, *Vol. 5*, 1586–1591.

Wehrli, F. W., D. Shaw, and J. B. Kneeland (1988). *Biomedical Magnetic Resonace Imaging.* VCH Publishers, Inc.

Wood, M. L. and R. M. Henkelman (1985). MR Image Artifacts from Periodic Motion. *Med. Phys., Vol. 12*, 143–151.

Wood, M. L. and R. M. Henkelman (1986). Suppression of Respiratory Motion Artifacts in Magnetic Resonance Imaging. *Med. Phys., Vol. 13*, 794–805.

Wood, M. L., M. J. Shivji and P. Stanchev (1995). Planar Motion Correction with Use of K-space Data Acquired in Fourier MR Imaging. *JMRI*, 57–65.

Xiang, Q. S. and R. M. Henkelman (1993). *K-Space* Description for MR Imaging of Dynamic Objects. *MRM 29*, 422–428.

Yang, W. F. and M. R. Smith (1998). Using a MRI Distortion Transfer Function to Characterize Motion of a Non-rigid Object. *IEEE Trans. on Medical Imaging (submitted).*

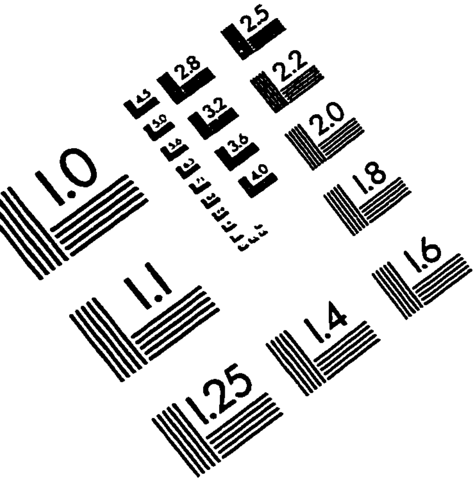
Zeng, J. and M. R. Smith (1994). Motion Artifact Removal using an ARMA Modeling Technique. *2nd SMR Meeting, San Francisco.*

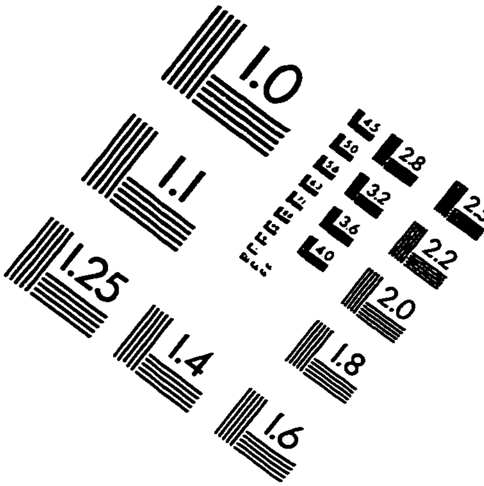
Zeng, J. X. (1996). *Quantitative Evaluation Methods for MRI Post-processing Algorithms, Master thesis.* Univ. of Calgary.

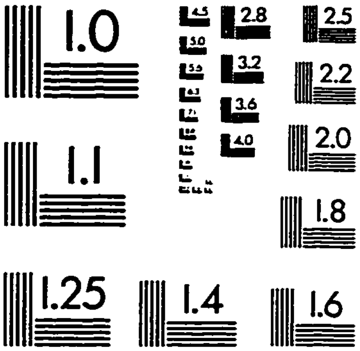
Zoroofi, R., Y. Sato, S. Tamura, H. Naito and L. Tang (1995). An Improved Method for MRI Artifact Correction due to Translational Motion in the Imaging Plane. *IEEE Trans. on Medical Imaging, Vol. 14*, 471–479.

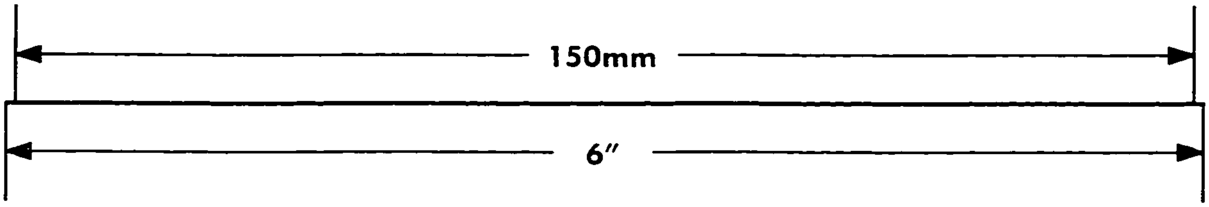
Zoroofi, R. A., Y. Sato, S. Tamura and H. Naito (1996). MRI Artifact Cancellation due to Rigid Motion in the Imaging Plane. *IEEE Trans. on Medical Imaging, Vol. 15*, 768–784.

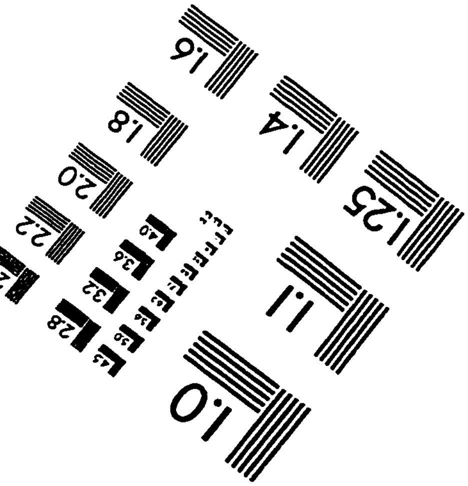
IMAGE EVALUATION TEST TARGET (QA-3)

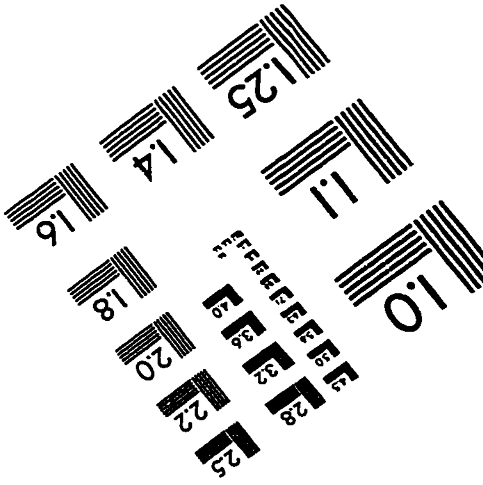












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989