

STRATEGIES FOR ADAPTIVE RADIATION THERAPY: ROBUST DEFORMABLE  
IMAGE REGISTRATION USING HIGH PERFORMANCE COMPUTING AND ITS  
CLINICAL APPLICATIONS

By

JUNYI XIA

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2009

© 2009 Junyi Xia

To my loving wife, *Yuming*, and to my parents  
for their unconditional love, support, and encouragement

## ACKNOWLEDGMENTS

First and foremost I would like to express my deepest appreciation and gratitude to my advisor, Dr. Sanjiv Samant, for his guidance throughout my years in UF. He provided me a friendly and welcome environment, gave me freedom to select my research directions, and spent countless hours in mentoring me. His support, patient, understanding, and kindness motivate me to do my best work.

I would also like to thank my co-chair, Dr. Yunmei Chen, for her invaluable ideas and suggestions. Her insight of mathematics makes me on the track and keeps me focused. It has really been honor working with her.

I would also like to thank other committee members, Dr. David Gilland, Dr. Wesley Bolch, and Dr. Murali Rao, for their useful discussions and suggestions.

It was very fortunate to work with my colleagues: Arun Gopal, Heeteak Chung, and Bart Lynch. We spent quality time together, discussing research problems as well as life experience. Special thanks would go to Dr. Chihray Liu and Jean Peng for their clinical insight.

Most importantly, I would like to thank my wife, Yuming, and my parents for their unconditional support and encouragement.

# TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS.....	4
LIST OF TABLES.....	7
LIST OF FIGURES .....	8
LIST OF ABBREVIATIONS.....	11
ABSTRACT.....	13
CHAPTER	
1 ROLE OF IMAGE REGISTRATION IN RADIATION THERAPY.....	15
1.1 Target Motion.....	20
1.2 Auto Re-contouring and Labeling.....	22
2 LITERATURE REVIEW OF DEFORMABLE IMAGE REGISTRATION .....	26
2.1 Review of Similarity Measures.....	27
2.1.1 Feature Based Measures.....	27
2.1.2 Intensity Based Measures.....	28
2.1.3 Hybrid Measures.....	30
2.2 Review of Transformation Models .....	31
2.2.1 Affine and Polynomial Models.....	31
2.2.2 Smooth Basis Function Models .....	32
2.2.3 Biomechanical Models.....	34
2.2.4 Optical Flow Models .....	35
2.2.5 Regularization on Dense Field Models .....	36
2.3 Review of Optimization Strategies .....	38
2.4 Review of Inverse Consistent Registration .....	39
3 VALIDATIONS OF DEFORMABLE IMAGE REGISTRATION .....	51
3.1 Accuracy .....	51
3.1.1 Physical Phantoms.....	52
3.1.2 Digital Phantoms.....	53
3.1.3 Validation Metrics .....	55
3.1.4 Visual Verification.....	58
3.2 Robustness .....	58
3.3 Consistency.....	59

4	A GENERALIZED INVERSE CONSISTENT CONSTRAINT FOR DEFORMABLE IMAGE REGISTRATION .....	61
4.1	The Inverse Consistent Constraint .....	62
4.2	The Inverse Consistent Demons (I-Demons) Algorithm.....	63
4.2.1	Demons Algorithm .....	63
4.2.2	Bijectivity Demons (B-Demons) Algorithm.....	64
4.2.3	I-Demons Algorithm.....	64
4.3	The Inverse Consistent Fast Diffusion Registration (I-FDR) Algorithm .....	65
4.4	A Robust Variational Model for Inverse Consistency .....	67
4.5	Algorithm Evaluations .....	69
4.5.1	Image Data for Evaluation .....	69
4.5.2	Evaluation Criteria.....	71
4.6	Results.....	72
4.6.1	Simulated Phantom Images.....	72
4.6.2	Physical Myocardium Phantom Images.....	74
4.6.3	Clinical Images .....	75
4.6.4	Comparison of FDR, I-FDR, and Christensen's Inverse Consistent Registration (CICR).....	77
4.7	Discussion and Conclusion.....	79
5	CLINICAL APPLICATION OF DEFORMABLE IMAGE REGISTRATION .....	105
5.1	Auto Re-contouring.....	105
5.1.1	Surface Reconstruction.....	108
5.1.2	Post Processing .....	108
5.2	Auto Internal Target Volume Generation.....	109
6	NEAR REAL-TIME DEFORMABLE IMAGE REGISTRATION USING GRAPHICS PROCESSING UNITS .....	117
6.1	Review of High Performance Computing In Medical Image Registration .....	118
6.1.1	Review of Image Registration Using FPGA.....	118
6.1.2	Review of Image Registration Using Cell Broadband Engine (CBE) .....	119
6.1.3	Review of Image Registration Using Supercomputers or Clusters .....	120
6.1.4	Review of Image Registration Using Graphics Processing Unit (GPU).....	123
6.2	Materials and Methods.....	128
6.3	Results.....	129
6.4	Discussion and Conclusion.....	132
	APPENDIX CODE SAMPLES OF RE-CONTOURING .....	140
	LIST OF REFERENCES .....	150
	BIOGRAPHICAL SKETCH .....	166

## LIST OF TABLES

<u>Table</u>	<u>page</u>
4-1 Change in step size of the FDR-group and Demons-group algorithms for robust measure .....	104
4-2 The mean and standard derivation (SD) of APE, ICE, and SSE of seven DIR algorithms using myocardium physical phantom, the image pixel dimension is 1 mm x 1 mm x 1 mm on X, Y, and Z direction.....	104
6-1 Maximum, mean and standard deviation of the difference in correlation coefficients between the CPU implementations and the GPU implementation in 100 iterations .....	138
6-2 Performance comparison between the CPU and GPU based implementations of DIR for clinical imaging data .....	138
6-3 Performance comparison between the CPU (2.8GHz Intel Dual Core with 1.5GB RAM) and GPU (NVidia 8800 GTX) implementations of the Demons DIR algorithm for clinical imaging data in Sharp et al.'s GPU implementation .....	139

## LIST OF FIGURES

<u>Figure</u>	<u>page</u>
1-1	Representative slices demonstrating 4DCT respiratory motion. Due to the diaphragm motion, the location of the lung tissue changes within the breathing cycle. .... 25
2-1	Deformable image registration algorithms consist of three components: similarity measures, transformation models, and optimization strategies ..... 48
2-2	1D parameter space illustrating the local minima (point B and point C on the blue curve) and the global minima (point A)..... 48
2-3	The forward transformation $h(x)$ matches the point A in the source image $S(x)$ to the point B in the target image $T(x)$ , the backward transformation $g(x)$ matches the point B in the target image $T(x)$ to the point A' in the source image $S(x)$ . .... 49
2-4	Computation of the inverse deformation field. The forward transformation $h(x)$ at point A determines the motion vector of point A, i.e, from point A to point B. .... 50
3-1	SPECT images of the dynamic physical phantom ..... 60
4-1	The simulated phantom..... 83
4-2	The evaluation of the FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma algorithms using the simulated phantom. .... 84
4-3	Change in energy with the number of iterations at a step size corresponding to the minimum APE ..... 85
4-4	The standard deviations (SD) of the FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma algorithms with the change in step size using the simulated phantom. .... 86
4-5	The evaluation of the Demons, I-Demons, and B-Demons algorithms using the simulated phantom ..... 87
4-6	The standard deviations (SD) of the Demons, I-Demons, and B-Demons algorithms with the change in step size using the simulated phantom. .... 88
4-7	The evaluation of the FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma algorithms using the physical myocardium phantom ..... 89
4-8	The standard deviations (SD) of the FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma algorithms with the change in step size using the physical myocardium phantom. .... 90
4-9	The evaluation of the Demons, I-Demons, and B-Demons algorithms using the physical myocardium phantom ..... 91

4-10	The standard deviations (SD) of the Demons, I-Demons, and B-Demons algorithms with the change in step size using the simulated phantom. ....	92
4-11	The evaluation of the FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma algorithms using the clinical 4DCT lung images .....	93
4-12	The evaluation of the Demons, I-Demons, and B-Demons algorithms using the clinical 4DCT lung images .....	94
4-13	The evaluation of the Demons-group and FDR-group algorithms using 4DCT clinical lung images for 200 iterations.....	95
4-14	Visual comparison of DIRs using 4DCT dataset of a lung cancer patient .....	96
4-15	Distribution of the inverse consistency error (ICE) in gray scale. Higher intensity region indicated the larger ICE. ....	97
4-16	Auto re-contouring of gross tumor volume (GTV).....	98
4-17	Comparison of the FDR, I-FDR, and CICR algorithms using the physical myocardium phantom. ....	99
4-18	Comparison of the FDR, I-FDR, and CICR algorithms using 4DCT images from a lung cancer patient .....	100
4-19	Comparison of two DIR algorithms using SSE and visual estimation .....	101
4-20	Quiver plot of the deformable field computed in Figure 4-19 after 600 iterations.....	102
4-21	The evaluation of the FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma algorithms using high noise myocardium phantom. ....	103
5-1	Flowchart of the auto re-contouring. The 3D surface mesh is reconstructed from the 2D layers of the manually contoured structures.....	111
5-2	Mesh deformation. The white cubes represent the nodes (A, B, and C) of a triangle from the planning surface mesh. ....	112
5-3	Surface reconstruction and surface mesh deformation. ....	113
5-4	A 1D example of interpolation based re-contouring and surface reconstruction based re-contouring. The contour in the middle plane (plane 2) is needed to be generated based on the known contours in the plane 1 and the plane 3. ....	114
5-5	Surface reconstruction from 2D planar contours .....	115
5-6	Auto re-contouring for lung image using DIR. Physician drawn contours (drawn on separate source CT) and computer generated contours, using auto re-contouring, have been overlaid on a target CT.....	115

5-7	ITV generation. The color map represents the probability that a GTV voxel occupies a specific pixel location for all phases in the respiratory cycle. The ITV is the union of all nonzero probability voxels.....	116
6-1	Memory architectures of the traditional parallel computers.....	134
6-2	Deformable image registration using lung imaging. 2D image slices presented here are based on volumetric deformable image registration using demon's method.....	135
6-3	Performance comparisons between the CPU (single threading and multithreading) and the GPU based implementations of the Demons DIR algorithm as a function of image size. ....	136
6-4	Computational efficiency of the GPU programming for DIR.....	137

## LIST OF ABBREVIATIONS

AP	Anterior/Posterior
API	Application Programming Interface
ART	Adaptive Radiation Therapy
CBCT	Cone Beam Computed Tomography
CBE	Cell Broadband Engine
CICR	Christensen's Inverse Consistent Registration
CT	Computed Tomography
CTV	Clinical Target Volume
DIR	Deformable Image Registration
EPID	Electronic Portal Imaging Device
FDR	Fast Diffusion Registration
FEM	Finite Element Method
FPGA	Field-Programmable Gated Array
GPU	Graphics Processing Unit
GTV	Gross Tumor Volume
HPC	High Performance Computing
ICC	Inverse Consistent Constraint
ICDIR	Inverse Consistent Deformable Image Registration
ICP	Iterative Closest Point
ICRU	International Commission on Radiation Units and Measurements
I-FDR	Inverse Consistent Fast Diffusion Registration
IGRT	Image-Guided Radiation Therapy
IMRT	Intensity-Modulated Radiation Therapy
ITV	Internal Target Volume

MRI	Magnetic Resonance Imaging
MRSI	Magnetic Resonance spectroscopic imaging
PCA	Principle Component Analysis
PDF	Probability Density Function
PPE	Power Processor Element
PSNR	Peak Signal-to-Noise Ratio
PTV	Planning Target Volume
RL	Right/Left
SI	Superior/Inferior
SPE	Synergistic Processor Element
SPECT	Single Photon Emission Computed Tomography
SD	Standard Deviation
SSD	Sum of Squared Distance
SSE	Sum of Squared Error
TPI	Time per iteration
TMPI	Time per mega-pixels per iteration

Abstract of Dissertation Presented to the Graduate School  
of the University of Florida in Partial Fulfillment of the  
Requirements for the Degree of Doctor of Philosophy

STRATEGIES FOR ADAPTIVE RADIATION THERAPY: ROBUST DEFORMABLE  
IMAGE REGISTRATION USING HIGH PERFORMANCE COMPUTING AND ITS  
CLINICAL APPLICATIONS

By

Junyi Xia

August 2009

Chair: Sanjiv Samant  
Cochair: Yunmei Chen  
Major: Nuclear Engineering Sciences

Image guided radiation therapy (IGRT) requires developing advanced methods for target localization. Once target motion is identified, the patient specific treatment margin can be incorporated into the treatment planning, accurately delivering the radiation dose to the target and minimizing the dose to the normal tissues. Deformable image registration (DIR) has become an indispensable tool to analyze target motion and measure physiological change by temporal imaging or time series volumetric imaging, such as four-dimensional computed tomography (4DCT). Current DIR algorithms suffer from inverse inconsistency, where the deformation mapping is not unique after switching the order of the images. Moreover, long computation time of current DIR implementation limits its clinical application to offline analysis.

This dissertation makes several major contributions: First, an inverse consistent constraint (ICC) is proposed to constrain the uniqueness of the correspondence between image pairs. The proposed ICC has the advantage of 1) improving registration accuracy and robustness, 2) not requiring explicitly computing the inverse of the deformation field, and 3) reducing the inverse consistency error (ICE). Moreover, a variational registration model, based on the maximum likelihood estimation, is proposed to accelerate the algorithm convergence and allow for inexact

image pixel matching within an optimized variation for noisy image pairs. The algorithm evaluation was carried out using a simulated phantom, a four-dimensional single photon emission computed tomography myocardium phantom, and clinical 4DCT images. After applying ICC, the ICE was reduced by up to 99% and the phantom error was reduced by up to 32%. For noisy image pairs, the likelihood based inverse consistent DIR algorithm outperformed other algorithms, it achieved fast convergence and attained a phantom error reduction of 56%, compared to the classic fast diffusion registration algorithm. Second, an auto re-contouring framework is developed for automatically propagate the planning contours in the planning image dataset to a new image dataset. It consists of DIR, surface reconstruction, and post processing. Visual estimation was applied to evaluate the re-contouring performance. The auto re-contouring framework is also applied to automatic generating internal target volume (ITV) and its probability density function by combing auto-contoured gross tumor volume of phase CTs. Third, a proof of concept study was carried out to accelerate the DIR computation using high performance graphics processing unit (GPU). It was demonstrated that the GPU implementation of DIR was able to speed up the computation by up to 60 times, achieved near real-time performance of DIR for clinical images (i.e., 1.8 seconds for image pairs with the size of 256 x 256 x 30 for 100 iterations) and improved the feasibility of applying deformable image registration in routine clinic procedures.

## CHAPTER 1 ROLE OF IMAGE REGISTRATION IN RADIATION THERAPY

Cancer is a group of disease characterized by growth and spread of tumors. In United States and Canada, more than 1,200,000 people are diagnosed with cancer annually. More than one in three people will develop cancer during their life time and about one in four will die of cancer. Among the individuals who developed cancer, about one half will be treated with radiation therapy, either alone or in combination with other cancer treatments. Radiation therapy is a clinical process using the ionizing radiation for the treatment of the malignant tumors (cancer). The goal of the radiation therapy is to deliver the radiation dose accurately to the well-defined target volume while sparing the radiation damage to the surrounding normal tissues.

Radiation therapy can be classified by the location of the radiation source:

1. External radiation therapy. The radiation is delivered from radioactive sources outside the body, such as linear accelerator;
2. Brachytherapy. The radiation is delivered from radioactive sources inside the body.

Radiation therapy is a complex procedure and involves many processes<sup>1</sup>, including 1) diagnosis and clinical evaluation; 2) therapeutic decision; 3) imaging for treatment planning; 4) target volume localization; 5) fabrication of treatment aids; 6) treatment simulation; 7) treatment planning; 8) treatment delivery; 9) patient evaluation; 10) patient follow-up. Not all patients will go through each step; however, it is very important to execute each step with the greatest accuracy possible. Errors or uncertainties occurred in one step will affect the accuracy of the subsequent steps and finally impact the patient outcome. Such closely related processes can also be interpolated as a chain. Among them, one crucial process is to determine the location and the extent of the diseases relative to the adjacent critical normal tissues (target volume localization). Target volume usually includes gross tumor volume (GTV), clinical target volume (CTV), and planning target volume (PTV). In ICRU report 52 and 60, GTV is defined as the gross

demonstrable extent and location of the tumor. Delineation of the GTV is possible if the tumor is visible, palpable or demonstrable through imaging. Clinical target volume consists of GTV and microscopic regions that cannot be detected by imaging techniques. Delineation of the CTV assumes that there are no tumor cells outside of the volume. Internal target volume extends the CTV with the compensation for internal physiological movements and variation in size, shape, and position of the CTV. Planning target volume (PTV) extends CTV and takes into account the patient motion and patient position errors.

The future of the radiation therapy will be focused on image guided radiation therapy (IGRT) and adaptive radiation therapy (ART). IGRT requires development of advanced hardware and software to identify the organ or the tumor distortion and motion. Once the tumor deformation is identified, the patient specific treatment margin can be incorporated into the treatment planning, in replace of the currently applied general margin. ART requires inclusion of imaging into the optimization of treatment planning on a dynamic basis. Target volume localization is, therefore, a very important step to the success of IGRT or ART. Target volume localization is a dynamic process because the volume might change during the course of the radiation therapy, for example, a maximum displacement of 7.5 mm was measured for the prostate during treatment <sup>2</sup>, average motion of 9.2 mm was observed in the lesions located near the heart <sup>3</sup>. Langen and Jones <sup>4</sup> classified the target motion into three categories: 1) patient position-related organ motion; 2) inter-fractional target motion; and 3) intra-fractional target motion. Patient position-related target motion is due to the change of the patient positioning, for example, when patient position changes from supine to lateral decubitus, relative location between the internal organs and the bone anatomies might change. Inter-fractional target motion occurs between the treatments fractions and it is on a day-to-day basis, intra-fractional target

motion may be due to patient weight gain or loss. Intra-fractional target motion occurs within the treatment when the radiation is delivered to the patients. Respiratory and cardiac motions are the main sources for intra-fractional target motion. Heinzerling et al.<sup>5</sup> reported a mean overall tumor motion of 13.6mm for the low lobe lung and the liver tumors. Furthermore, the target motion pattern could change during the radiation therapy, for example, the lung cancer patient might be more nervous for the first few fractions than the later fractions, and the change of their breathing pattern could affect the magnitude of the target motion. The widely applied intensity-modulated radiation therapy (IMRT) techniques can deliver radiation to the target volumes of complex shape and minimize the dose to the nearby normal tissues. Therefore it is especially important to understand the target motion for IMRT because high dose gradient is located near the target area, even small variation of the target volume or the target location might significantly change the planning dose distribution. For the past few years, the target motion has been intensively studied. In the following, we will give a brief review of target location techniques using Electronic portal imaging devices (EPID), optical devices, electromagnetic treatment positioning device, and four-dimensional computed tomography (4DCT).

EPID was introduced to monitor the target motion by locating the implanted radio-opaque markers. EPID has been under the development by the researchers and linear accelerator manufactures for almost 20 years, from the initial television camera-based system to current flat panel systems. EPID was widely used in the patient positioning. Herman et al.<sup>6</sup> investigated the patient setup accuracy of the prostate cancer patients by aligning the orthogonal portal image with the simulated digital reconstructed radiographs, the mean misalignment of the markers was 5.6 mm after initial patient setup for 20 patients. Similar studies were carried out by other researchers<sup>7-10</sup>. However, implanting markers is an invasive surgical procedure and is not always

feasible for all cancer patients, for example, marker implantation will be complex for certain lung cancer patients. Moreover, portal imaging only provides two-dimensional (2D) geometric information while target volume change is happened in three-dimensional (3D) space. It is difficult to predict the 3D tumor motion from the 2D projections. Finally, the number of markers used in the previous study was limited, usually consisting of 3 to 5 gold markers, which are not enough to represent the full details of the target motion.

Optical tracking is another technique to measure target motion<sup>11</sup>. In optical tracking, infrared markers are first attached to the patient. The infrared markers are either active or passive. Active markers are infrared light-emitting diodes. Passive markers are generally sphere or disks coated with reflective material to reflect the infrared light. The infrared detectors, such as charged coupled device (CCD) camera, detect the reflected light from the infrared markers and determine the markers' 3D locations from the collected 2D images. Optical tracking system is a non-invasive technique; it has the advantage of high spatial resolution and providing real-time patient tracking. However, current optical systems are limited to track external markers. In order to predict the internal target motion, the correlation between internal motion and external marker motion has to be modeled. Gierga et al.<sup>12</sup> investigated the motion correlation between the external marks and internal markers and concluded that relative large discrepancy occurred between the external marker motion and internal marker motion even though they are generally correlated in respiratory motion case. Furthermore, optical tracking system is difficult to reveal the tumor shape evolution during the course of the radiation therapy.

An electromagnetic treatment target positioning device, called Calypso 4D localization system ( Calypso Medical Technologies, Seattle, WA), was undergo clinical evaluation in several institutions<sup>13</sup>. The Calypso system consists of beacon transponder, electromagnetic

array, tracking stations, and optical system. Beacon transponder contains a miniature electronic circuit and has a size of 8.7 mm in length and 1.85 mm in diameter. The transponder does not have internal energy source. When excited by the electromagnetic field generated by the electromagnetic array, the transponder resonates and emits a signal at a unique frequency, which is acquired by the array and processed by the software to determine the location of the transponder. The optical system is used to continuously track the position of the array and determine the transponder location with respect to the linear accelerator's isocenter. Even though the Calypso system provides real-time 3D target information, it has the following disadvantages<sup>14</sup>: 1) it is an invasive techniques and beacon transponders are needed to be surgically implanted, 2) transponders migrant within the tissue and the stability is an concern. Studies found the mean distance variation between transponders was 1.3 mm at 4 days after implant and was 0.8mm at 14 days after implant.

Recently 4DCT has been introduced to clinical applications. 4DCT images are reconstructed by time-resolved 3D images and are capable of presenting both spatial and temporal patient anatomical information. For example, in 4DCT lung imaging, all CT projection images are first acquired while patient breathe freely on the CT table. Then the image projections corresponding to specific breathing phases are binned together using the information from breathing control devices (such as spirometer or active breathing control (Varian Medical Systems, Palo Alto, California)). Figure 1-1 shows a 4DCT images consisting of a sequence of ten equally divided phases of three-dimensional CT images over the patient's breathing cycle. Phase 50 corresponds to the maximum expiration; phase 0 corresponds to the maximum inspiration. The duration of expiration was acquired between phase 0 and phase 50, the duration of inspiration was acquired between phase 50 and phase 0. With the acquired time series 3D

images, image registration can be used to study and measure the target motion and patient anatomical change. Beside 4DCT, image registration can also be used to measure the target motion from cone beam CT (CBCT) images. CBCT makes it possible to acquire high resolution volumetric 3D images to determine the patient setup errors prior to irradiation. CBCT utilizes large area collimation of x-rays, and involves scattered radiation which results in loss of image quality and increases imaging dose to patients compared with the narrow beam collimation in conventional computed tomography (CT). CBCT is widely used to facilitate high precision target positioning<sup>15-19</sup>.

By definition, image registration computes the transformation between two image volumes to make them similar. It can be classified into rigid image registration and deformable image registration (DIR). In rigid image registration, the transformation is limited to translation, rotation, scaling, and shearing. The rigid image registration ignores the possible target deformation that occurs naturally in various organs, i.e. lung and prostate, and is only useful when the organ deformation is small. In contrast, DIR computes the deformation fields that can be used to measure the target deformation. A detailed review of the DIR was discussed in Chapter 2. In the following, we will give a brief review of the role of DIR in the frame of radiation therapy.

## **1.1 Target Motion**

Fraction to fraction variations of patient anatomy and patient setup lead to the uncertainty of the dose delivery, such as, under-dose to the tumor and over-dose to the healthy tissues. The dose uncertainties might have more significant effects on IMRT, where regions of high dose and/or high dose gradient exist.

One of the clinical applications of DIR in target motion is to measure the respiratory motion. Coselmon et al.<sup>20</sup> proposed applying the DIR consisting of mutual information and thin-

plate splines to measure the lung motion of 11 patients, 30 control points were manually selected near the interior of the lung and the border of the lung. The result indicated that landmarks motion was  $-0.4 \text{ mm} \pm 2.7 \text{ mm}$  in the RL direction,  $8.1 \text{ mm} \pm 6.6 \text{ mm}$  in the AP direction, and  $3.2 \text{ mm} \pm 8.6 \text{ mm}$  in the IS direction. Liu et al.<sup>21</sup> investigated the tumor motion from 152 lung cancer patients using the “accelerated demons”<sup>22</sup>, the tumor motion of 2.6 mm to 14.7 mm was found in the SI direction, -1.4 mm to 5.8 mm in the LR direction, -1.9 mm to 2.7 mm in the AP direction. Christensen et al.<sup>23</sup> modeled the lung tissue as elastic material and applied inverse consistent registration algorithm to measure the correlation of the motion vector and the readings of the spirometer, the result indicated the strong correlation between the average expansion/compression of the lung measured by image registration and air flow rate measured by spirometry.

Measuring liver tumor motion is another application of DIR. Researches show that live tumor motion could reach 19 mm in tidal breathing<sup>24</sup>. A finite element modal based multi-organ DIR method, MORFEUS, was developed by Brock et al.<sup>25</sup>. It described the surface interfaces between the organs, the material properties were assigned to each organ for allowing accurate deformation of internal structures. For five liver cancer patients<sup>26</sup>, it was found that up to 1 cm motion between the diaphragm and the tumor center of mass for registration of the exhale and inhale CTs, up to 6.8 mm motion was observed between the CT and the cone-beam CT. Using gated magnetic resonance (MR) imaging, Rohlfing et al.<sup>27</sup> analyzed the liver motion using an intensity based approach including rigid registration for global motion and B-spline free-form DIR for local deformation. 3D MR images of four volunteers were acquired between end-of-inspiration and end-of-expiration. An average of 10 mm deformation over the entire liver was observed in four volunteers by the deformation field.

## 1.2 Auto Re-contouring and Labeling

The DIR computes the correspondence between the pixels in the two image volumes. This correspondence enables us to auto re-contouring and labeling. There are generally two frameworks for re-contouring<sup>28</sup>: 1) registration based framework and 2) segmentation based framework. The registration based framework can be grouped into four steps:

1. Deformation fields computing. The deformation between the planning volume (the one has contoured structures) and the target volume (the one needed to be contoured).
2. 3D surface mesh generated from 2D contours. Typically the outlined structures are in 2D and it is desirable to convert the series of 2D geometry into 3D surface mesh in order to apply 3D deformation fields.
3. Structure propagation. Each point on the surface mesh is moved to the new location by the deformation fields.
4. Post processing. The deformed 3D structures are re-sliced into new 2D structures for visualization.

The segmentation based framework applies iterative algorithms to seek the distinct features in the images, such as the high gradient area, as the boundary. Compared with registration based framework, the segmentation based framework does not affected by the accuracy of the physician-drawn contours; however, it cannot be used to re-contouring the structures whose boundaries are not obvious, such as CTV or PTV. In contrast, registration based framework is able to propagate all the physician-drawn contours. It has to be noted that the quality of the propagated contours using DIR based approaches relies on not only the accuracy of the DIR algorithm but also the accuracy of the original contours.

Collins et al.<sup>29</sup> presented a brain auto re-contouring method based on the brain atlas and the non-linear registration procedure using MRI. Lu et al.<sup>28</sup> proposed a method for automatic contours propagating using 4DCT images. His method combined the DIR and surface reconstruction. Surface meshes are constructed from the manually contoured structures, after

deformed by the deformation field, the deformed surface meshes are re-sliced to the 2D contours.

The details of auto re-contouring will be discussed in Chapter 5.

As DIR gains focus in the radiation therapy, it faces challenges. The challenges of DIR are

1. **Long computation time.** Because of solving a large amount of variables, DIR is computational intensive and requires long computation time. Typical reported CPU times in published literatures for DIR of CT data sets include: 5 minutes for a 128 x 128 x 128 volume using demons<sup>30</sup>, 3 minutes for a 256 x 256 x 61 volume on Pentium III 933MHz PC using calculus of variations<sup>31</sup>, 37 seconds for 1024 x 1024 (2D) on SGI OCTANE 175MHz using calculus of variations<sup>32</sup>, 6 minutes for a 256 x 256 x 61 volume on Pentium 2.8GHz PC using accelerated demons<sup>22</sup>. The long computation time limits its applications to be widely used in the clinics. To the best of our knowledge, no current commercial treatment planning system is able to performance deformable registration in near real-time.
2. **Validation.** Online applications of DIR in radiotherapy demand high accuracy. Due to the complexity of the DIR, gold standard of the DIR is difficult to obtain. Studies have been carried out using a simple mathematical deformation for algorithm validation<sup>31</sup>, however, the simple mathematical deformation is not sufficient to model the tumor motion or deformation actually occurred in the patient.
3. **Lacking physiological constraints.** DIR assumes one-to-one mapping, which indicates that the pixels correspondence within two image volumes should be invariant of the registration sequence, for example, the correspondence between the pixels in image A and image B should be independent of whether deforming image A to image B or image B to image A. This relation is also called inverse consistency. The current DIR algorithms used radiation therapy usually does not always preserve inverse consistent because of lack of constraints in matching criteria to uniquely describe the one-to-one correspondence between the images<sup>33</sup>.
4. **Robustness.** The robustness of the registration algorithm involves registration results should not significantly changed when parameters vary, in other words, the registration result should be insensitive to parameter selections.

The rest of the dissertation has been organized into the followings: Chapter 2 provided a literature review of the DIR including inverse consistent registration; Chapter 3 provided a survey of the existing validation techniques for DIR; Chapter 4 discussed a novel inverse consistency constraints for constraining DIR; Chapter 5 present the clinical application of auto re-contouring using DIR and auto internal target volume generation; In Chapter 6, a proof-of-

concept study was developed to achieve near real time DIR using emerging technology of graphics process units.

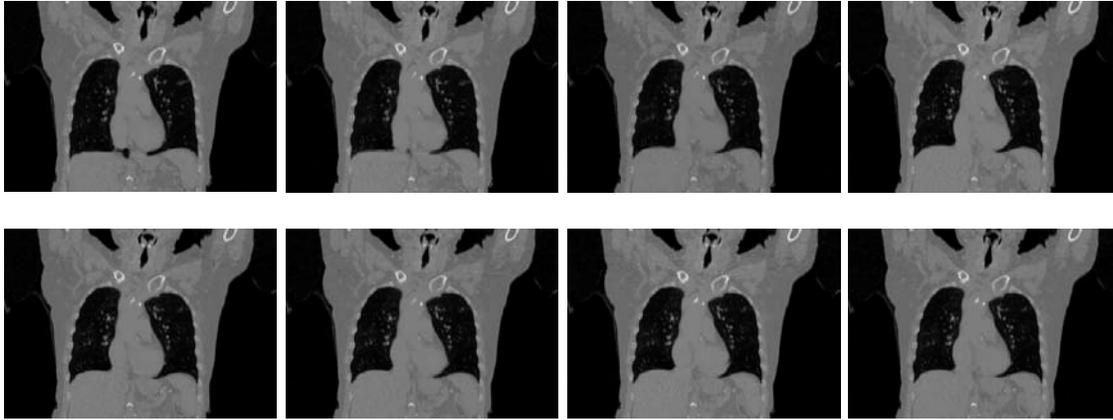


Figure 1-1. Representative slices demonstrating 4DCT respiratory motion. Due to the diaphragm motion, the location of the lung tissue changes within the breathing cycle.

## CHAPTER 2 LITERATURE REVIEW OF DEFORMABLE IMAGE REGISTRATION

Image registration is to find the transformation that aligns two or more image volumes together. It is a fundamental problem in medical imaging and it can be applied within or between different imaging modalities, such as computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), and ultrasound (US). The goal of image registration in radiation therapy is to reduce the treatment margins, allow performing safe dose escalation, and improve the outcome of patient treatment <sup>34</sup>.

Image registration methods have been extensively reviewed <sup>35-43</sup>. It can be categorized by the type of the transformations involved, i.e., rigid registration versus deformable registration (also called non-rigid). For rigid registration, only translation and rotation are considered, for instance, 3D rigid registration has totally six parameters: three rotations and three translations. Automatic rigid inter-modality and intra-modality image registrations have been developed for years and are available in some commercial software. The rigid registration can be applied to the cases where the relative shape change of the target (i.e., tumor or organ) is small. However, in reality, most of the human organ motions are not always modeled by the rigid transformation <sup>44</sup>. To describe more complex change of the target motion, DIR is introduced to compute the transformation with more degree of freedoms. Reviews of the DIR algorithms can be found in <sup>40, 43, 45, 46</sup>.

Generally, DIR algorithms consist of three components (Figure 2-1): similarity measures, transformation models, and optimization strategies. Similarity measures compute the similarity between two images. Transformation models specify how to deform one image to match another. Optimization strategies seek to determine the parameters of the transformation model in order to maximize or minimize the matching criterion.

In the following, we will first give a review of each component of DIR, and then we will review the inverse consistent registrations, which reduce the asymmetry of the common unidirectional DIR algorithms.

## **2.1 Review of Similarity Measures**

In DIR, similarity measures can be categorized into three groups: 1) Feature (geometric) based measures, including points, curves, and surface. Incorporating those features ensures that the transformation is biologically valid and is consistent with the underlying anatomical and physiological constraints. 2) Intensity based measures. Image similarity is defined as mathematical or statistical measurements of the intensity distribution of the two images. Intensity based measures usually fully automatic and need less human intervention. 3) Hybrid measures. It combines both the feature based measures and intensity based measures.

### **2.1.1 Feature Based Measures**

In feature based measures, points (or landmarks)<sup>47-51</sup>, curves, and surfaces are used for registration purpose. Both anatomical features (such as bifurcations) and fiducial markers can be used as points or landmarks. In general, there are two steps in point based registration: 1) detecting the landmarks and 2) establishing the correspondence of the landmarks. Landmarks can be either manually selected by experts or automatically detected using image processing algorithms<sup>52</sup>. The measurement of the landmarks localization error was essential for accurate registration<sup>50</sup>. Many studies devoted to investigate how to establish the robust correspondence between the landmarks, for example, Besl and McKay<sup>53</sup> introduced the iterative closest point (ICP) algorithm to efficiently locate the corresponding points between two images without shape representation. The ICP algorithm iteratively searches the closest point between two point sets and estimates motion using the corresponding point pairs. ICP is very popular for registration of two point sets, however, it was susceptible to gross outliers. Many extensions of ICP method

were proposed to improve the robustness on outlier rejection, such as random sampling and the least median of squares estimator<sup>54</sup>, hybrid ICP algorithm<sup>55</sup>, Gaussian-weighted correspondence matrix<sup>56</sup>, and Hausdorff distance<sup>57</sup>. Correspondence of the landmarks can be determined by the mass and stiffness matrix based on the Gaussian of the distances<sup>58</sup>. Cross and Hancock<sup>59</sup> built a graph representation to constrain the search of the correspondences by Delaunay triangulation and expectation-maximization optimization algorithm.

Besides points or landmarks, curve features (such as ridge or crest lines) extracted from the images can be used for deformable registration. Crest lines are defined as joining points of zero maximal principle curvature. In brain imaging, Subsol et al.<sup>60</sup> first represented the gyri and sulci of different volunteers with the crest lines, then registered them to construct the brain atlas. Curve features were also used in the multimodality registration, where the anatomies revealed similar geometric features in both imaging modality. Maintz<sup>61</sup> registered CT and MRI images of human head using ridge extraction. Other image registrations using ridge features<sup>62-64</sup> were also proposed.

### **2.1.2 Intensity Based Measures**

Comparing with the feature based measures, intensity based measures do not require feature extraction; instead, they utilize mathematical or statistical criteria to match the images. Based on types of image modalities, intensity based similarity measures can be grouped into two categories: 1) measures for mono-modality and 2) measures for multi-modality.

#### **2.1.2.1 Measures for mono-modality**

The assumption of mono-modality measures is that the pixel intensity of images does not change during the deformation, or intensity change is linear. An example of the former is sum of squared distance (SSD) or sum of squared error (SSE), an example of the later is correlation coefficient (CC). SSD and CC can be written in Equation (2.1) and (2.2), respectively.

$$SSD = \frac{1}{N} \sum_{i=1}^N (I_1(x_i) - I_2(x_i))^2 \quad (2.1)$$

$$CC = \frac{\sum_{i=1}^N [(I_1(x_i) - \bar{I}_1) \cdot (I_2(x_i) - \bar{I}_2)]}{\sqrt{\sum_{i=1}^N (I_1(x_i) - \bar{I}_1)^2 \cdot \sum_{i=1}^N (I_2(x_i) - \bar{I}_2)^2}} \quad (2.2)$$

where  $\bar{I}_1$  and  $\bar{I}_2$  are the mean pixel intensity of image  $I_1$  and image  $I_2$ , image  $I_1$  and image  $I_2$  have  $N$  pixels. The first application of SSD was reported by Horn and Schunck<sup>65</sup> for solving an energy function incorporating optical flow and linear differentiable regularization. Lu et al.<sup>31</sup> and Fisher et al.<sup>32</sup> proposed using SSD as a similarity measure in their energy functions and applied different numerical schemes to obtain the deformation field. In Christensen and Johnson's work<sup>66</sup>, SSD was utilized to measure the deformation in the brain MRI images.

### 2.1.2.2 Measures for multi-modality

For multi-modality images, directly comparing pixel intensity is limited because same object may have different pixel intensities from different modalities. For example, the bony anatomies have high pixel intensities in CT images; however, they have low pixel intensities in MRI images. One approach to measure the similarity for multi-modal images is to find a function  $f$  that after applying the function, the intensity between  $I_1$  and  $I_2$  are similar, that is,  $I_1 \approx f(I_2)$ .

An example of these measures is correlation ratio  $\eta(I_1|I_2)$ <sup>67</sup>

$$\eta(I_1 | I_2) = 1 - \frac{Var[I_1 - E(I_1 | I_2)]}{Var(I_1)} \quad (2.3)$$

where  $Var(I_1)$  is the variance of  $I_1$ ,  $E(I_1 | I_2)$  is the conditional expectation,  $Var[I_1 - E(I_1 | I_2)]$  measures the degree of  $I_1$  that is functionally independent of  $I_2$ .  $\eta$  is in the range between 0 and 1,  $\eta=0$  indicates that  $I_1$  and  $I_2$  are functional independence;  $\eta=1$  indicates  $I_1$  and  $I_2$  are

functional dependence. Two images are considered to have more functional dependence if their similarity increases.

Another approach for similarity measures is to analyze probability relationship between images, such as mutual information (MI). Basically, MI indicates how well one image can represent another. MI is defined as

$$MI(I_1, I_2) = H(I_1) + H(I_2) - H(I_1, I_2) \quad (2.4)$$

where  $H(I_1)$  and  $H(I_2)$  are the Shannon entropy of image  $I_1$  and  $I_2$ ,  $H(I_1, I_2)$  is joint entropy of image  $I_1$  and  $I_2$ . Applications of using mutual information in multi-modality DIR were reported in cardiology using 3D ultrasound<sup>68</sup>, brain<sup>69</sup>, and liver<sup>70</sup>.

### 2.1.3 Hybrid Measures

The intensity based similarity measures utilizes the intensity information to match the images without segmenting the image features, in contrast, the feature based registration matches the images based on the features that usually sparsely distributed in the image. To achieve more robust and accurate measures, hybrid measures combine both intensity based measures and feature based measures together. Christensen et al.<sup>71</sup> proposed a two-step method to recover the large organ deformation due to the intracavitary applicators. Firstly, a landmark based fluid registration was applied to obtain the initial deformation, and then volumetric viscous fluid registration utilized the initial deformation to compute the final deformation. The hybrid approach has the advantage of robustness from land-maker based algorithms and flexibility from intensity based methods. Pluim et al.<sup>72</sup> reported registration robustness improvement after combining mutual information and spatial gradient features. Hellier and Barillot<sup>73</sup> proposed a framework incorporating intensity based SSD similarity measures with cortical constraints for deformable inter-subject brain registration. The cortical constraints were expressed as the

extracted cortical sulci segmentation. Cachier et al. <sup>74</sup> proposed an iconic feature based registration incorporating intensity based similarity measures with geometric distance.

## 2.2 Review of Transformation Models

Transformation models determine how to deform one image to match another image.

Transformation models can be classified by different criteria, such as the effective range of the transformation <sup>34</sup>. Here we will classify them by the types of the transformations, which include 1) affine/polynomial models, 2) smooth basis function models, 3) biomechanical models, 4) optical flow models, and 5) regularization on dense field models. Affine and polynomial models, smooth basis function models require control points and they act like interpolation function to find the displacement of non-grid locations given the displacement on the grids. For others transformations, such as biomechanical models, optical flow models, and regularization on dense field models, control points are not required.

### 2.2.1 Affine and Polynomial Models

The affine transform deals with linear transformations, i.e., rotation, translation, shearing, and scaling. It preserves the collinearity, i.e., parallel lines are parallel after transformation.

Affine transformation can be expressed as

$$T(x) = Ax + b \quad (2.5)$$

where  $A$  is the transformation matrix,  $b$  is the translation vector,  $x$  and  $T(x)$  are the input and

output vector, respectively. In 2D space,  $A = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$ . It is, however, difficult to

represent a non-linear deformation using only one single affine transformation. Collins et al. <sup>75</sup> proposed combining a group of local affine transformations to compute the brain deformation.

Transformations can also be represented by polynomial functions <sup>76</sup>, which are defined as

$$T(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (2.6)$$

where  $a_0, a_1, a_2,$  and  $a_n$  are the polynomial coefficients,  $n$  is the degree of the polynomial function. One drawback of polynomial model is that oscillations tend to appear for high-order polynomial and local deformation is difficult to represent <sup>40</sup>.

### 2.2.2 Smooth Basis Function Models

The summation of smooth basis functions, such as thin-plate spline <sup>77</sup> and B-spline, is another approach to model transformations. The name thin-plate spline refers to the physical analogy of bending a thin sheet metal. Thin-plate spline is defined as

$$T(x, y) = A_1 + A_2x + A_3y + \sum_{i=1}^N F_i r_i^2 \ln r_i^2 \quad (2.7)$$

where  $r_i^2 = (x - x_i)^2 + (y - y_i)^2 + d^2$ ,  $(x_i, y_i)$  is a control point in thin-plate spline. From physics point of view, Equation (2.7) describes a plate of infinite surface deformation under the loads centered at  $(x_i, y_i)$ , where the deflation is  $T_i$ . The distribution of the load is specified by  $d^2$ . The load becomes a point load when  $d^2$  approaches zero. Equation (2.7) contains  $N+3$  unknowns ( $A_1, A_2, A_3,$  and  $F_i (i=1, \dots, N)$ ).  $N$  equations can be obtained by substituting point  $(x_i, y_i)$  and deflation  $T_i$  into Equation (2.7), the rest three equations can be obtained using

$$\sum_{i=1}^N F_i = 0 \quad (2.8)$$

$$\sum_{i=1}^N x_i F_i = 0 \quad (2.9)$$

$$\sum_{i=1}^N y_i F_i = 0 \quad (2.10)$$

Equation (2.8) assumes the plate under the loads will not move, and remain stationary. Both Equation (2.9) and (2.10) indicate the moments along  $x$  and  $y$  axis are zero, that is, no rotation is allowed for the plates. Thin-plate spline was first applied in image registration of remote sensing

images by Geshtasby<sup>78</sup>, thin-plate spline was also used for MRI/MRSI mapping<sup>79</sup>, inverse consistent registration<sup>80</sup>, and analyzing shape variation in schizophrenia<sup>81</sup>. Similar to thin-plate spline, multiquadric function is another type of basis functions, it is defined as

$$T(x, y) = F_1 + F_2x + F_3y + \sum_{i=4}^N F_i R_i(x, y) \quad (2.11)$$

where  $R_i(x, y) = \sqrt{(x-x_i)^2 + (y-y_i)^2 + d^2}$ ,  $(x_i, y_i)$  is the control point,  $d$  has the same meaning as in Equation (2.7). Equation (2.11) contains  $N$  unknowns, which are determined by substituting point  $(x_i, y_i)$  and deflation  $T_i$  into Equation (2.11). Both thin-plate spline and multiquadric models were found to be suitable when the number of control points was not large and the distance variation between the control points was small<sup>82</sup>. The main disadvantage of thin-plate spline and multiquadric models is that each control point has a global influence of the transformation, any change of one control point will affect the transformation and these global effects of control points make it difficult to model complex and local deformation.

Rather using thin-plate spline models, some researchers used B-spline models by adjusting the locations of control points. Using B-spline models<sup>83</sup>, the deformation field can be described as

$$T(x, y, z) = \sum_{l=0}^3 \sum_{m=0}^3 \sum_{n=0}^3 B_l(u) B_m(v) B_n(w) \phi_{i+l, j+m, k+n} \quad (2.12)$$

where  $\phi_{i,j,k}$  is the control point in a control points mesh with the size of  $n_x \times n_y \times n_z$  and uniform spacing  $\delta$ ,  $i = [x/n_x] - 1$ ,  $j = [y/n_y] - 1$ ,  $k = [z/n_z] - 1$ ,  $u = x/n_x - [x/n_x]$ ,  $v = y/n_y - [y/n_y]$ ,  $w = z/n_z - [z/n_z]$ ,  $B_l$  represents the  $l$ th B-spline basis function

$$\begin{aligned}
B_0(u) &= (1-u)^3 / 6 \\
B_1(u) &= (3u^3 - 6u^2 + 4) / 6 \\
B_2(u) &= (-3u^3 + 3u^2 + 3u + 1) / 6 \\
B_3(u) &= u^3 / 6
\end{aligned}
\tag{2.13}$$

The advantage of B-spline models is that they are locally controlled, changes of control point locations only affect the transformation of the neighborhood of the control points, which make B-spline computationally efficient<sup>83</sup>. However, the disadvantages of traditional B-spline registration are 1) certain measures have to be utilized to prevent folding of the deformation field<sup>43</sup>; 2) the challenge of determining optimal numbers of control points<sup>84</sup>, few control points leads to coarse match and large number of control points may lead to local oscillations. Application of B-spline models included measuring liver motion<sup>27</sup>, PET-CT image fusion<sup>85</sup>, cardiac modeling<sup>86,87</sup>, and breast deformation<sup>83</sup>.

### 2.2.3 Biomechanical Models

Biomechanical models use physical properties of organs and its boundary conditions to compute deformations and motions, such as brain shift<sup>88,89</sup>, brain tumor deformation<sup>90</sup>, breast deformation<sup>91,92</sup>, heart motion<sup>93</sup>, and liver motion<sup>26,94</sup>. It is based on the Finite Element Method (FEM). The FEM is a classical engineering analysis technique that produces solutions to partial differential equations (PDEs) describing complex spatially distributed systems and processes. It has been widely used in structural and continuum mechanics. In essence, the FEM strategy divides the domain of interest (e.g. liver) into an interconnected set of sub-regions or elements by tetrahedral mesh. Discrete approximations to the PDEs that govern the physical processes to be simulated (e.g. consolidation theory) are developed on each element which can possess its own local properties, allowing complex geometries and tissue heterogeneities to be represented through a simple building-block structure. Given sufficient resolution of the

geometry of interest, FEM methods produce highly accurate solutions to complex equations under realistic conditions.

In order to simulate organ deformation, some FEM parameters should be considered:

1) Boundary condition. The accuracy of the finite element model depends on the accuracy of the boundary condition. Zhang et al.<sup>95</sup> reported applying contact-impact analysis on the boundary of the lung surface between inhale and exhale breathing phases to improve the accuracy.

2) Material property. The physical properties of each organ should be specified, such as Young's modulus and Poisson's ration. However those physical properties usually vary between patients. Chi et al.<sup>96</sup> investigated the registration accuracy with respect to the change of the physical properties for rectum wall, bladder wall, and prostate.

#### 2.2.4 Optical Flow Models

The optical flow was introduced by Horn and Schunck<sup>65</sup> to estimate the motion between frames in an image sequence. The fundamental assumption is that the brightness of the images is preserved, that is, the pixel intensity of the same object does not change within two image frames. Given two images  $I(x, y, z, t)$  and  $I(x + \delta x, y + \delta y, z + \delta z, t + \delta t)$ , the optical flow velocity  $u$  can be described as

$$I(x, y, z, t) = I(x + \delta x, y + \delta y, z + \delta z, t + \delta t) \quad (2.14)$$

Equation (2.14) can be rewritten as

$$\frac{\partial I(x, y, z, t)}{\partial t} = 0 \Rightarrow \frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial z} \frac{dz}{dt} + \frac{\partial I}{\partial t} = 0 \quad (2.15)$$

Equation (2.15) can be simplified as

$$\nabla I \cdot u = -\frac{\partial I}{\partial t} \quad (2.16)$$

where  $u = [dx/dt, dy/dt, dz/dt]$ . Equation (2.16) is under constrained and many smoothing constraints were proposed to address this problem. Thiron<sup>30</sup> proposed an iterative method to constrain the flow velocity  $u$ . At each iteration, the flow velocity  $u$  was firstly computed without constraints, and then the Gaussian filter was applied to convolve  $u$  to enforce the smoothness. Carefully choosing the smoothing parameters, such as the standard deviation and the width of the Gaussian filter, is important to obtain good registration results. Horn and Schunck<sup>65</sup> proposed a constraint by minimizing the square of the magnitude of the gradient of the optical flow velocity  $u$ . Hellier et al.<sup>97</sup> used robust estimators as a constrain to preserve the natural discontinuity of the deformation field, which usually appear on the structure boundary.

### 2.2.5 Regularization on Dense Field Models

DIR was also formulated as an optimization problem. The cost function (also called energy function)  $E$  usually contains similarity measure ( $E_{sim}$ ) and regularization ( $E_{reg}$ ). Similarity measure describes how similar source image  $S(x)$  and target image  $T(x)$  is, regularization is to constrain the deformation field. The goal is to find the optimal deformation field  $u$  to minimize the energy function  $E(u)$ ,

$$\begin{aligned} E(u) &= E_{sim}(u) + \lambda E_{reg}(u) \\ u &= \arg \min E(u) \end{aligned} \tag{2.17}$$

where  $\lambda$  is the weighting factor for balancing the similarity term and regularization term. Here smaller value of similarity term indicates better matching. For certain similarity measures, such as cross correlation, where larger value indicates better matching, the similarity term in Equation (2.17) can be represented by the negative of the similarity measures. Equation (2.17) can be explained in a Bayesian framework<sup>46,98</sup>: the similarity measures term acts like a likelihood term expressing the probabilities of matching between the source image and target image; the regularization term acts like a prior expressing the prior knowledge of the deformation fields.

Another way to understand Equation (2.17) is to consider the similarity term as a driving force to maximizing the similarity between two images, to consider regularization term as a penalty function to constrain the transformation.

Both feature based similarity measures and intensity based similarity measures can be applied as similarity term, readers can refer to Chapter 2.1 for the detail discussion of the possible forms of the similarity measures. For regularization term, the common one is the linear differential regularization, where large variation in deformation field is significantly penalized.

$$E_{reg}(u) = \|\nabla u\|^2 \quad (2.18)$$

Another group of regularizations is based on physical models. Linear elastic regularization<sup>66, 99</sup> assumes the deformation is governed by the Navier equation in elasticity theory and deformed in an isotropic and homogeneous elastic entity, it can be written as

$$E_{reg}(u) = \mu \nabla^2 u + (\lambda + \mu) \nabla(\nabla \cdot u) \quad (2.19)$$

where  $\nabla^2$  is the Laplacian operator,  $\nabla$  is the Nabla operator,  $\nabla \cdot u$  is the divergence of  $u$ ,  $\lambda$  is the Lamé constants,  $\mu$  is the elastic material property. The elastic regularization assumes small deformation. To handle large deformation, viscous fluid regularization<sup>100-102</sup> was introduction. Viscous fluid regularization constrains the deformation as the flow of the viscous fluid; it can be efficiently expressed as

$$\begin{aligned} & \mu \nabla^2 v + (\lambda + \mu) \nabla(\nabla \cdot v) \\ v &= \frac{\partial u}{\partial t} + v \cdot \nabla u \end{aligned} \quad (2.20)$$

where  $v$  is the velocity of the deformation. Because viscous fluid regularization allows large deformation, the deformation field is needed to be checked to avoid folding. Other regularizations include bi-harmonic<sup>103</sup> model and membrane<sup>104</sup> model.

### 2.3 Review of Optimization Strategies

After constructing a cost function, the next step is to obtain the transformation parameters using optimization. This problem can be stated as: given a cost function  $f$  and the unknown parameters, find the optimal set of parameters that maximize or minimize the cost function. Generally whether maximizing or minimizing the cost function is trivial because maximizing  $f$  equals to minimizing  $-f$  and vice versa. Many registration algorithms are suitable using existing optimization strategies, such as Powell's method, steepest gradient descent method, conjugate gradient method, and downhill simplex method. A review of optimization algorithms can be found in the classic book by Press<sup>105</sup>. We consider the case for steepest descent optimization algorithms. Let  $u$  the deformation field to be optimized,  $f(u)$  is the energy function,  $\nabla f(u)$  is the first derivatives of the energy function, an optimized  $u$  can be computed using the steepest descent optimization algorithm

$$\frac{\Delta u}{\Delta t} = -\nabla f(u) \quad (2.21)$$

where  $\Delta u$  is the increment of the deformation field,  $\Delta t$  is the step size. Equation (2.21) shows that the optimal direction is always the direction of the local downhill gradient ( $-\nabla f(u)$ ).

One of the difficulties with optimization algorithms in DIR is that the optimized transformation parameters that results in a good images may not be physically meaningful<sup>43</sup>. A common example is the change of the topology, such as tearing or folding of the deformation field. In reality, both tearing and folding do not occur in the evolution of anatomical structures. However, tearing may result in optimized transformation parameters that make two images more similar to each other, i.e., having lower cost function value, even though the resulting transformation is not physically valid. Therefore, relying on the cost function itself might not be sufficient to achieve good optimization parameters. Care must be taken to make sure the

resulting transformation parameters are physically feasible. For example, a common method to check the preservation of topology is to compute the determinant of the Jacobian matrix of the deformation field. Positive determinant indicates the topology is preserved.

Another challenge optimizers facing is “local minima”, where the optimizer is converged to the local minimum and the optimized parameters do not reflect the global minima. For example, in Figure 2-2, we consider optimizing a parameter with the cost function in blue,  $X$  axis is the parameter space. The point  $A$  is the ideal global minimum the optimization algorithm should find, however, it is very common for optimization algorithms to converge in the local minima, i.e., point  $B$  or point  $C$  in Figure 2-2. One of the approaches of avoiding the local minima is to construct an alternative convex cost function, such as the red curve in Figure 2-2. Unfortunately, parameters space of cost function cannot always be easily constructed as a convex function, especially for deformation image registrations that contain possibly thousands of parameters. Multi-resolution is another widely used approach to reduce the possibility of trapping in the local minima, however, its convergence to the global minima is not guaranteed.

## 2.4 Review of Inverse Consistent Registration

The goal of image registration is to find the correspondence between two images, which are called source image  $S$  and target image  $T$ . Here source image  $S$  is deformed to match target image  $T$ . The correspondence between two images are often assumed to be one-to-one, indicating each point in template image  $T$  is only mapped to one point in source image  $S$ . Inverse consistency is defined as the correspondence between two images is invariant when switching the order of the target image and source image, for example, as shown in Figure 2-3, point  $A$  in image  $S$  is mapped to point  $B$  in image  $T$  when deforming image  $S$  to image  $T$ ; for inverse consistent registration, after switching order of  $T$  and  $S$ , point  $B$  in image  $T$  should still be mapped to point  $A$  in image  $S$ , i.e., point  $A'$  and point  $A$  are at the same location. Let us define

the forward transformation  $h(x)$  the transformation from source image  $S$  to target image  $T$ , and the backward transformation  $g(x)$  the transformation from target image  $T$  to source image  $S$ , mathematically an inverse consistent transformation can be expressed by

$$h(g(x)) = I \text{ or } g(h(x)) = I \quad (2.22)$$

where  $h(x) = x + u(x)$ ,  $g(x) = x + v(x)$ ,  $u(x)$  and  $v(x)$  are the forward and the backward deformation field, respectively. Nielsen and Markussen<sup>106</sup> argued that inverse consistency was one of the most important properties of deformation fields. It should be noted that the concept of inverse consistency is different than the concept of the inverse of deformation field. For inverse consistency, both forward and backward transformations are involved. For inverse of the deformation field, only one transformation, such as forward transformation or backward transformation, is required.

Because some inverse consistency methods require computing the inverse of the deformation field, let us first give a brief review of how to compute inverse. Figure 2-4 illustrates the problem of computing inverse of the deformation field. One of the common misunderstandings of computing inverse is to simply add a negative sign to the forward transformation, i.e.,  $\overline{DA}$  is the inverse of  $\overline{AB}$  in Figure 2-4, this method might lead to the wrong inverse because the true inverse vector is  $\overline{CA}$ , not  $\overline{DA}$ . Methods of computing the inverse of the deformation field had been reported<sup>66, 107-109</sup>. Christensen and Johnson<sup>66</sup> proposed an iterative scheme to calculate inverse deformation field based on grid search, a good initial guess of the inverse was desired for fast convergence. Cachier and Rey<sup>108</sup> computed the inverse using the Newton scheme. Ashburner<sup>109</sup> reported a method based on the concept of tetrahedral mesh and affine transformation inversion, where the deformation field was decomposed into local affine transforms and the inverse deformation field was computed by combining the inversion of local

affine transformation. Chen et al. <sup>107</sup> reported an iterative inverse computation approach based on the fixed-point theory to achieve fast convergence. The Insight Registration and Segmentation Toolkit <sup>110</sup> included an inverse deformation field computation approach using a look-up table based scatter data interpolation, however, unlike iterative approaches, look-up table based approaches requires large memory and long computation time.

The unidirectional DIR algorithms have the difficulty of maintaining the inverse consistency because of the lack of matching criteria to uniquely define the one-to-one correspondence between the images <sup>66</sup>. Cachier et al. <sup>108</sup> suggested the asymmetry ( inverse inconsistency) of unidirectional registration was due to 1) order non-preservation of the energies. For two forward transformation  $h_1$  and  $h_2$ , where energy function  $E(h_1) > E(h_2)$ , their inverse energy  $g_1$  and  $g_2$  might not follow  $E(g_1) > E(g_2)$ , 2) non-stable transformation space, 3) local minima. The optimization might reach different local minima of the energy when the target image and the source image were switched.

There are a number of works on developing inverse consistent DIR. One category of inverse consistent methods is to minimizing coupled symmetric energy functions involving forward and backward transformation, each of them consists of a similarity term, a regularization term, and an inverse consistency term. Christensen and Johnson <sup>66</sup> was among the first ones to introduce the inverse consistent image registration by jointly estimating the forward and backward transformation using the symmetric energy function. The proposed energy function was given as

$$\begin{aligned}
E(h, g) &= \sigma E_{sim} + \rho E_{reg} + \chi E_{icc} \\
E_{sim} &= \int_{\Omega} \|S(h(x)) - T(x)\|^2 + \|S - T(g(x))\|^2 dx \\
E_{reg} &= \int_{\Omega} \|Lu(x)\|^2 + \|Lw(x)\|^2 dx \\
E_{icc} &= \int_{\Omega} \|h(x) - g^{-1}(x)\|^2 + \|g(x) - h^{-1}(x)\|^2 dx = \int_{\Omega} \|u(x) - \tilde{w}(x)\|^2 + \|w(x) - \tilde{u}(x)\|^2 dx \\
h(x) &= x + u(x); \quad g(x) = x + w(x)
\end{aligned} \tag{2.23}$$

where  $T$  and  $S$  are the target and the source image,  $h$  and  $g$  are the forward transformation and the backward transformation,  $u$  and  $w$  are the forward and the backward displacement field,  $\tilde{u}(x)$  and  $\tilde{w}(x)$  are the inverse of the forward and the inverse of the backward displacement field,  $\sigma$ ,  $\rho$ , and  $\chi$  are the weighting factors,  $L$  is the linear elastic regularization operator

$$Lu(x) = -\alpha \nabla^2 u(x) - \beta \nabla(\nabla \cdot u(x)) + \gamma u(x), \text{ where } \nabla = [\partial/\partial x_1, \partial/\partial x_2, \partial/\partial x_3] \text{ and}$$

$\nabla^2 = [\partial^2/\partial x_1^2, \partial/\partial x_2^2, \partial/\partial x_3^2]$ . To solve Equation (2.23), 3D Fourier series representation was used to parameterize the forward and backward displacement field. The displacement field has the form

$$\begin{aligned}
u(x) &= \sum_k \mu[k] e^{j\langle x, N\theta[k] \rangle} \\
w(x) &= \sum_k \eta[k] e^{j\langle x, N\theta[k] \rangle}
\end{aligned} \tag{2.24}$$

where  $\langle x, N\theta[k] \rangle = 2\pi k_1 x_1 + 2\pi k_2 x_2 + 2\pi k_3 x_3$ ,  $\mu[k]$  and  $\eta[k]$  are (3x1) complex-valued vectors. After Fourier parameterization, the problem of computing  $u(x)$  and  $w(x)$  is equivalent to finding their corresponding  $\mu[k]$  and  $\eta[k]$ . The advantage of using Fourier parameterization is that the number of the parameters could be reduced. This model gave considerably good result with carefully chosen parameters. The limitation of this model is that the inverse deformation field is required to be explicitly computed at each iteration, which may introduce cumulative numerical error and result in long computation time. Similar inverse consistent regularization was also applied to consistent B-spline registration<sup>111</sup>, consistent landmark and intensity based

registration using thin-plate spline<sup>80, 112</sup>, and local large deformation<sup>113</sup>. To improve the computational efficiency, Christensen et al.<sup>114</sup> reported consistent registration on the object of interest, incorporating information of both boundary and the intensity inside the object, rather than the entire image. Yeung and Shi<sup>115, 116</sup> proposed an inverse consistent approach considering stochastic uncertainty

$$\begin{aligned}
(h + \gamma_h) \circ (g + \gamma_g) &= I + r \\
\gamma_h &= |h - g^{-1}| \\
\gamma_g &= |g - h^{-1}|
\end{aligned} \tag{2.25}$$

where  $\gamma_h$  and  $\gamma_g$  are stochastic error for forward transformation  $h$  and backward transformation  $g$ ,  $r$  is the error due to the imperfectness of the inverse consistency. A generalized total least square approach was adopted to solve the stochastic problem.

Zhang et al.<sup>69</sup> presented a consistent multi-modality deformable by estimating the forward transform  $h$  and backward transform  $g$  in a variational framework. The consistent energy function was expressed as

$$\begin{aligned}
E &= E_f + E_b \\
E_f &= Sim(S(h), T) + \lambda_1 \int_{\Omega} \varphi(D(h(x))) dx + \lambda_2 \int_{\Omega} (g \circ h - Id(x))^2 dx \\
E_b &= Sim(T(g), S) + \lambda_1 \int_{\Omega} \varphi(D(g(x))) dx + \lambda_2 \int_{\Omega} (h \circ g - Id(x))^2 dx
\end{aligned} \tag{2.26}$$

where  $E_f$  and  $E_b$  are the forward and backward registration energy.  $Sim(S(h), T)$  and  $Sim(T(g), S)$  are mutual information.  $\varphi(D(h(x)))$  and  $\varphi(D(g(x)))$  are the regularization term,

$\int_{\Omega} (g \circ h - Id(x))^2 dx$  and  $\int_{\Omega} (h \circ g - Id(x))^2 dx$  are the inverse consistency residue, which are minimized when forward transformation  $g$  and backward transformation  $h$  are inverse to each other. Alvarez et al.<sup>117</sup> proposed a symmetrical energy function to estimate optical flow with

occlusion detection. His model replaced the similarity term and regularization term in Equation (2.26) with

$$E_{sim} = Sim(S(h), T) = \frac{1}{\max \|\nabla T(x)\|^2} \int_{\Omega} \|S(h(x)) - T(x)\|^2 dx \quad (2.27)$$

$$E_{reg} = \int_{\Omega} trace(\nabla(h)^T D(\nabla T(x)) \nabla(h)) dx$$

where  $\nabla(h)$  is the Jacobian matrix of  $h(x)$ ,  $D(\nabla T(x))$  is a projection matrix in perpendicular direction to  $\nabla T(x)$ . The idea of applying Equation (2.27) is to prevent the displacement field being smoothed across the object boundary.

Leow et al.<sup>118</sup> proposed an consistent unidirectional approach by implicitly including inverse consistency constraint to directly solve the following energy

$$E(h, g) = \underbrace{\int_{\Omega} |S(h(x)) - T(x)|^2 dx}_{E_1} + \underbrace{\int_{\Omega} |T(g(x)) - S(x)|^2 + \lambda R(g(x))}_{E_2} \quad (2.28)$$

The forward and backward transformation can be numerical updated as

$$h^{n+1} = h^n + \varepsilon(\eta_1 + \eta_2), \quad g^{n+1} = g^n + \varepsilon(\xi_1 + \xi_2) \quad (2.29)$$

where  $\varepsilon$  is a small number,  $\eta_1$  and  $\xi_1$  are vector field representing gradient descent direction of  $E_1$  in the forward and backward direction accordingly,  $\eta_2$  and  $\xi_2$  are similarly defined for  $E_2$ .  $\eta_1$  and  $\xi_2$  can be computed by

$$\begin{aligned} \eta_1 &= -\nabla_h E_1(h) \\ \xi_2 &= -\nabla_g E_2(g) \end{aligned} \quad (2.30)$$

$\eta_2$  and  $\xi_1$  can be determined from the inverse consistency relationship between  $h^{n+1}$  and  $g^{n+1}$

$$\begin{aligned} h^{n+1} \circ g^{n+1} &= id \\ (h^n + \varepsilon(\eta_1 + \eta_2)) \circ (g^n + \varepsilon(\xi_1 + \xi_2)) &= id \end{aligned} \quad (2.31)$$

Taking Taylor's expansion of Equation (2.31) and collecting up to the first order term, one gets

$$\eta_1 + \eta_2 = -D(h)[\xi_1(h) + \xi_2(h)] \quad (2.32)$$

where  $D$  is the Jacobian operator, it is expressed as

$$D = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

where  $x_1, \dots, x_n$  are the directions of the deformation vectors,  $f_1, \dots, f_n$  are the component of the deformation field in each direction. Using Equation (2.32),  $\eta_2$  and  $\xi_1$  can be determined by

$$\begin{aligned} \eta_2 &= -D(h)\xi_2(h) \\ \xi_1 &= -D(g)\eta_1(g) \end{aligned} \quad (2.33)$$

Therefore, the update scheme of  $h^{n+1}$  can be obtained by Equation (2.29) , Equation (2.30) , and Equation (2.33). More important,  $h^{n+1}$  can be updated using only previous forward transformation  $h^n$  without computing the backward transformation  $g$  and inverse field  $g^{-1}$ .

However, one disadvantage of this model is that the truncation error in deriving Equation (2.32) using Taylor expansion may not be small and may be accumulated during the iterations.

Cachier et al.<sup>108</sup> investigated the asymmetric results of uni-directional registrations and proposed an inversion-invariant energy to symmetrize the asymmetric energy function. In their method, the energy function (Equation (2.34)) consisted of symmetric similarity term and symmetric regularization term. Equation (2.34) could be solved using two approaches: 1) alternatively solving both forward and backward mapping using either finite element implementation or finite difference implementation. 2) Using finite element implementation to solve Equation (2.35), only involved computing forward mapping by approximating backward transformation using forward transformation

$$\begin{aligned}
E_f(T, S, u) &= E_{sim} + E_{reg} = \int_{\Omega} \|T(x + u(x)) - S(x)\|^2 dx + \int_{\Omega} \|\nabla u(x)\|^2 dx \\
E_b(S, T, w) &= E_{sim} + E_{reg} = \int_{\Omega} \|S(x + w(x)) - T(x)\|^2 dx + \int_{\Omega} \|\nabla w(x)\|^2 dx
\end{aligned} \tag{2.34}$$

$$E_f(T, S, u) = \frac{1}{2} \left[ \int (1 + |\nabla u|) (T(x + u(x)) - S(x))^2 + \lambda \left(1 + \frac{1}{|\nabla u|}\right) \|\nabla u\|^2 \right] \tag{2.35}$$

Inverse consistency can also be achieved in the context of large deformation diffeomorphic flow based registration. By definition, a diffeomorphic transformation is a mapping with the properties of one-to-one, onto, and continuously differentiable. Avants et al.<sup>119</sup> presented a inverse consistent approach by dividing the transformation into two parts, forward transformation  $h_{0,t}$  and backward transformation  $g_{1,1-t}$ . Assume there exists an intermedian image  $I(x)$  that is within the geodesic path of connecting source image  $S(x)$  and target image  $T(x)$ ,  $h_{0,t}$  is the transformation from  $S(x)$  to  $I(x)$  at time  $t$  and  $g_{1,1-t}$  is the transformation from  $T(x)$  to  $I(x)$  at time  $t$ . The cost function is given by

$$\begin{aligned}
E &= \int_0^{0.5} \|v_t^h\|^2 + \|v_t^g\|^2 + \lambda \|S(h_{0,t}) - T(g_{1,1-t})\|^2 dt \\
\text{subject to } v_{0.5}^h &= v_{0.5}^g, \quad \|v_t^h\|^2 = \|v_{1-t}^g\|^2
\end{aligned} \tag{2.36}$$

where  $v_t^h$  and  $v_t^g$  are flow satisfying  $\frac{dh_{0,t}}{dt} = v_t^h$  and  $\frac{dg_{1,t}}{dt} = v_t^g$ . Equation (2.36) can be explained as deforming both  $S(x)$  and  $T(x)$  toward the ‘‘mean’’ image at time  $t=0.5$  to obtain a solution by minimization with respect to  $h_{0,t}$  and  $g_{1,1-t}$ . The complete forward transformation  $\phi_h$  and backward transformation  $\phi_g$  can be expressed by

$$\phi_h = h_{0,t} \circ g_{1,1-t}^{-1} \quad \text{and} \quad \phi_g = g_{1,1-t} \circ h_{0,t}^{-1} \tag{2.37}$$

Equation (2.37) indicates the transformation is inverse consistent because  $\phi_h \circ \phi_g = id$  and  $\phi_g \circ \phi_h = id$ . Similar approaches were also proposed by Beg and Khan<sup>120</sup>, Joshi et al.<sup>121</sup>, and Yang et al.<sup>122</sup>.

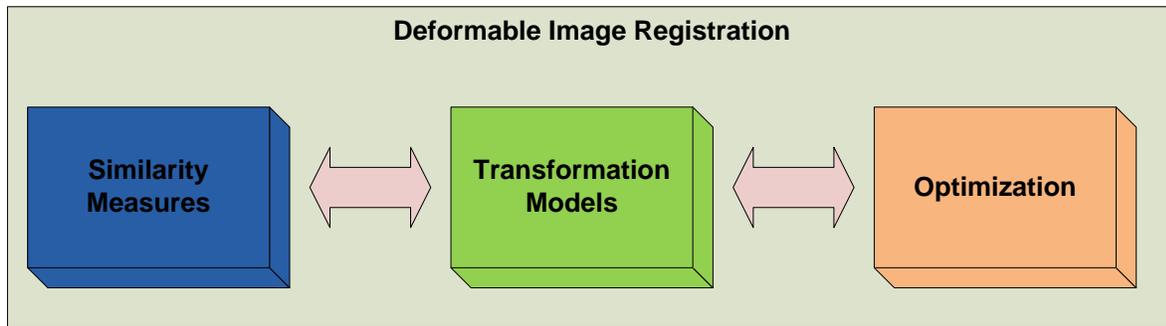


Figure 2-1. Deformable image registration algorithms consist of three components: similarity measures, transformation models, and optimization strategies.

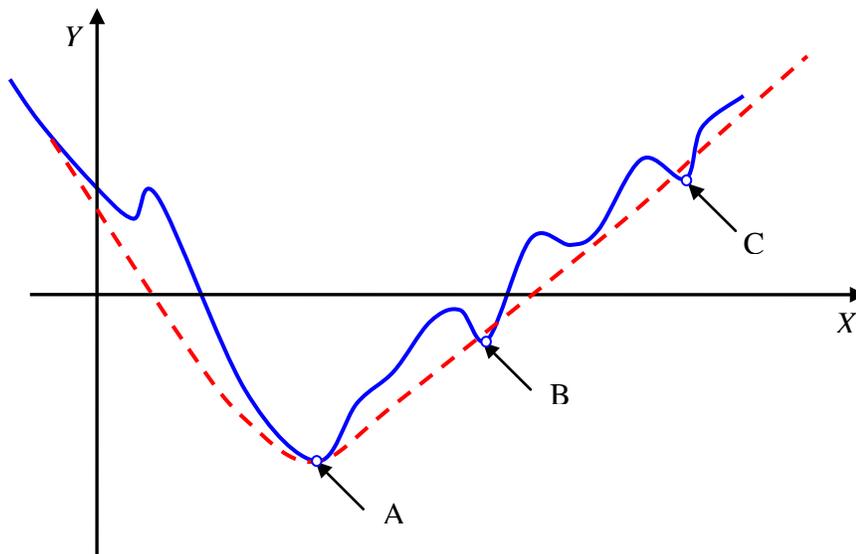


Figure 2-2. 1D parameter space illustrating the local minima (point B and point C on the blue curve) and the global minima (point A). X axis is the parameter space; Y axis is the value of the cost function, showing in blue curve. The goal of the optimization algorithm is to find the parameters corresponding to the global minima and avoid local minima. The dotted red curve demonstrating the ideal convex parameter space, which optimization algorithm will always converge to the global minima.

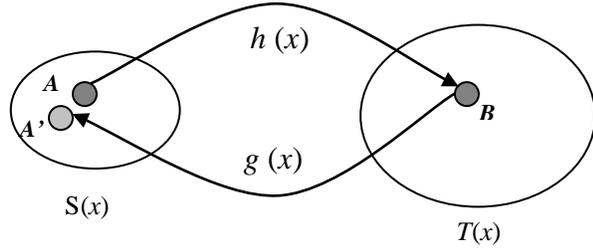


Figure 2-3. The forward transformation  $h(x)$  matches the point  $A$  in the source image  $S(x)$  to the point  $B$  in the target image  $T(x)$ , the backward transformation  $g(x)$  matches the point  $B$  in the target image  $T(x)$  to the point  $A'$  in the source image  $S(x)$ . Ideally the choice of registration direction (i.e., forward transformation or backward transformation) should not affect the correspondence, i.e.,  $A$  and  $A'$  are at the same point. However, this is usually not hold for unidirectional registration algorithms.

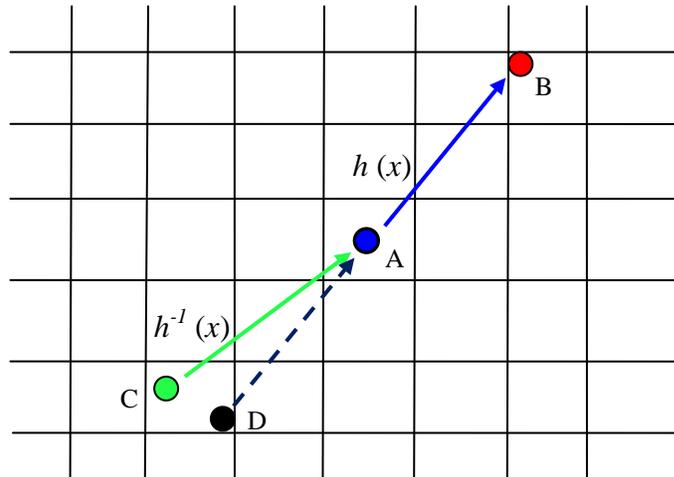


Figure 2-4. Computation of the inverse deformation field. The forward transformation  $h(x)$  at point A determines the motion vector of point A, i.e, from point A to point B. Assuming  $h(x)$  is known, to compute inverse transformation  $h^{-1}(x)$  at point A is to find the point C, which moves to point A by the motion vector. Point D is the point determined by simply adding a negative sign to the forward motion vector  $\overline{AB}$ . Point C and point D are not always the same.

## CHAPTER 3 VALIDATIONS OF DEFORMABLE IMAGE REGISTRATION

Validation of DIR refers to demonstrating a registration algorithm applied to representative data in a given application consistently succeeds within the acceptable range (e.g., maximum or mean error) for the application<sup>43</sup>. In DIR, validation can be challenge because ground truth, also called gold standard, is not generally available. The general validation approaches applied in the rigid registration validation, such as landmarks registration errors, is not enough for validating the points further away from the landmarks. Jannin et al.<sup>123</sup> suggested registration validation may include the following criteria: 1) accuracy; 2) precision; 3) robustness; 4) consistency or closed loop. In this chapter, we will discuss registration validation from the view of accuracy, robustness, and consistency. Unlike human studies, precision is not a big validation issue in image registration because most algorithm hardly contain random components and will produce consistent result once the image data and registration parameters are determined. It has to be noted that validation through precision, robustness, and consistency do not require the existence of gold standards.

### 3.1 Accuracy

Accuracy is the degree to which a measurement is true or correct. Deformable registration requires high accuracy to represent the actual deformation for its clinic applications. The most straight forward accuracy measurements is to compute the difference between the measured values with the true values, however, as mentioned before, it is very difficult to obtain the ground truth from the deformation in the real world application. A trade-off is to simulate the deformation by either mathematical simulation or physical phantom study. For patient images, validation metrics and its corresponding mathematical or statistical tools had been studied for the validation purpose.

### 3.1.1 Physical Phantoms

Kashani et al.<sup>124</sup> investigated the feasibility of using a simple deformable phantom to validate deformable registration algorithms. The phantom consisted of a diagnostic thoracic imaging phantom with a deformable foam insert, where small plastic markers were distributed. The foam was deformed by a one-dimension drive stage in SI direction to simulate the exhale and the inhale respiratory state. The markers were served as ground truth and were digitally removed before the registration. The maximum errors of markers' position were reported between 7.5 mm and 9 mm for three registration algorithms, which were thin-plate splines, B-splines, and affine DIR algorithms. Another deformable lung phantom was proposed by Serban et al.<sup>125</sup>. The phantom consisted of a piston driving mechanism and a Lucite cylinder filled with water and a latex balloon stuffed with dampened natural sponges, where the sponges represented lung tissues, nylon wires and Lucite bead represented vascular and bronchial bifurcations. Lu et al.<sup>31</sup> reported a gel-balloon deformable phantom, where the deformation was simulated by inserting or removal heavy oil to inflate or deflate the balloon.

Parker and Gilland<sup>126</sup> evaluated the myocardial wall motion for gated cardiac emission tomography using a modified commercial available, dynamic, physical phantom (Figure 3-1). It consisted of a water-filled torso containing two lungs, a Teflon spine, and a beating heart. The heart assembly was composed of a 50 mL latex membrane and a 100 mL cylindrical latex membrane. The myocardium motion was monitored by measuring the radioactivity injected into the volume between the two membranes. The computer controlled pump managed the beating motion while sending ECG signal to the SPECT for gating purpose. Ten point source markers were attached to the outer surface of the myocardium with approximate uniform spacing. Marker scan was first carried out to locate the markers position. In marker scan, each marker was injected approximately 3 mCi of <sup>99m</sup>Tc, after that, the phantom was scanned with three

consecutive 60 minutes gated SPECT using low energy-high resolution collimator. The markers were then allowed to decay to negligible level. The markers locations were computing using center of mass for SPECT images. To simulate the myocardium motion in the myocardium scan, high activity  $^{99m}\text{Tc}$  (73 mCi) was injected into the myocardium compartment with a long SPECT scan time. The phantom was not moved during the marker scan and myocardium scan in order to ensure the physical correspondence between the markers and myocardium images. The myocardium phantom has the advantage of “automatic” marker removal by the marker scan and the consecutively scan to represent the myocardium motion, without the need to manually remove the markers from the image volume that might change the pixel intensity distribution in the images.

The advantages of physical phantoms are that the motions of the markers represent the real physical motion in the phantom and the motion is physiologically consistent. The disadvantages of the physical phantoms are 1) the difficulty to design a phantom considering all the variability encountered in the clinical situations. All existing physical phantoms use simple materials, such as sponges, membrane, or gels, to represent their targeting organs, which can be different than the reality. 2) The validation using physical phantoms is not on voxel-by-voxel basis and overall accuracy could be limited by the number of markers available in the phantom. Physical phantoms have difficulty to estimate the registration errors beyond the markers locations.

### **3.1.2 Digital Phantoms**

Digital phantom utilizes mathematical, statistical, or image processing methods to simulate the organ deformation. Gholipour et al.<sup>127</sup> presented a digital brain phantom to validate the registration between low resolution images of echo planar imaging (EPI) and high resolution images of proton density image (PD). To simulate the artifacts and problems of EPI images, the digital phantom was constructed as followings: first, a high resolution PD images was down

sampled and the interested regions were segmented, then the effect of signal loss artifacts was added to the interested regions in down sampled PD images using the simplified signal fade model, after that both deterministic and random distortions were simulated using markers incorporating physical measured distortion and random number generator, finally a thin-plate spline interpolation was used to generate the deformation field. Because of the usage of thin-plate spline interpolation, this digital brain phantom might be bias toward the thin-plate spline based DIR.

Commonly used mathematical functions include harmonic functions<sup>31, 107</sup>, sinusoidal function<sup>30</sup>. Harmonic functions are defined as a twice continuously differentiable function  $f$  satisfying Laplace equation, i.e.

$$\nabla^2 f = 0 \text{ where } \nabla^2 = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \dots + \frac{\partial^2}{\partial x_n^2}$$

The following harmonic functions<sup>31, 107</sup> were applied to simulate the 2D motion:

$$\begin{aligned} u(x) &= bx \cos(m\theta) \\ u(y) &= by \cos(m\theta) \\ \theta &= \tan^{-1}(y/x) \end{aligned} \tag{3.1}$$

where  $u(x)$  and  $u(y)$  are the simulated displacement field on  $x$  and  $y$  direction, respectively.

Compared to physical phantoms, deformation generated by digital phantoms allows evaluating deformation fields using voxel-by-voxel comparison, however, the disadvantage of mathematical functions is that 1) the generated deformation may be too simple and not reproduce the true anatomical deformations of the patients; 2) there might exist functional dependencies between the models for data simulation and the models for evaluation<sup>123</sup>. To generate clinically realistic deformation field, Wang et al.<sup>22</sup> applied the fiducial marker based thin-plate spline registration to register a pair of CT images and considered the registration result as the gold

standard for validating other registration methods. This approach is more clinically realistic than the simple mathematical deformation, however, the validation result may be more biased toward the thin-plate spline registration. To reduce the bias of the deformation field generated from a particular deformation algorithm, Xue et al.<sup>128</sup> described a statistical approach to generate the deformation field from a number of training samples for inter-individual and intra-individual brain deformation. The idea was to estimate the probability density function (pdf) of the deformation fields from a limited number of training samples. In this approach, the global and local training deformation was statically analyzed using Wavelet-Packet transform, Jacobian matrix, and Markov random field. The simulated deformation can then be constructed by randomly sampling the statistical distribution in an unconstrained or a landmark-constrained fashion. Biomechanical model was also used to generate simulated deformation field. Fahmi et al.<sup>129</sup> proposed to simulate 2D kidney deformation and 3D brain deformation using finite element method. The generated deformation field was considered as gold standard for validation purpose.

Overall, the realism of the simulated deformation is difficult to prove and the simulated deformation may not take into account the true viability encountered in clinic situations.

### 3.1.3 Validation Metrics

The validation metrics refers to the mathematical or statistical methods to measure the outcome of the registration. Validation metrics can be based on pixel intensity distribution, volumetric criteria, and distance measures.

1. Pixel intensity distribution, such as correlation coefficient (CC)<sup>130</sup>, sum of squared error (SSE), mutual information. CC is defined as

$$CC = \frac{\sum (T - \bar{T})(S - \bar{S})}{\sqrt{\sum (T - \bar{T})^2 \sum (S - \bar{S})^2}} \quad (3.2)$$

where  $T$  and  $S$  are the target and the source image respectively,  $\bar{T}$  and  $\bar{S}$  represent the mean intensity of the target image and the source image, respectively. Larger CC value indicates better similarity of two images. The perfect match between two images gives a CC value equals 1. CC measurement is limited to the same modality images and is not sensitive to the global intensity scale change. SSE is defined as

$$SSE = \sum_{i=1}^n (T_i - S_i)^2 \quad (3.3)$$

where  $T_i$  and  $S_i$  are the target and the source image, respectively. CC can be used in the cases where the pixel intensities in the images are linearly related, in contrast, SSE can only be used in the cases where the range of the pixel intensities is same.

Shannon introduced the ideas of mutual information in 1948. Mutual information measures how much information one random variable tells about another random variable. In the field of image registration, it is a measure of how well one image matches with the other because mutual information is larger for more similar images and smaller for less similar images. The main advantage of using MI is that the actual form of the intensity dependency between the two images does not have to be specified. This makes MI well suited as a criterion of multi-modality registration. The mutual information  $I(A, B)$  can be expressed as

$$I(A, B) = H(A) + H(B) - H(A, B) \quad (3.4)$$

where  $H(A)$  and  $H(B)$  are the Shannon entropy of image  $A$  and  $B$ ,  $H(A, B)$  is the joint entropy of image  $A$  and  $B$ . Moreover, the normalized mutual information (NMI)

$$NMI(A, B) = \frac{H(A) + H(B)}{H(A, B)}$$

has been shown to be considerably more robust than mutual

information.

2. Volumetric criteria, such as dice similarity index (DSI)<sup>130, 131</sup>, average volume<sup>132</sup>, intensity variance metric<sup>131</sup>, determinant of Jacobian matrix<sup>133</sup>. The Dice similarity index is the overlap ratio between the volume of interest in the deformed image and the gold standard, it can be written as

$$DSI = \frac{2 \times |V_c \cap V_g|}{|V_c| + |V_g|} \quad (3.5)$$

where  $V_c$  is the volume of interest of the deformed image,  $V_g$  is the gold standard. For a perfect match,  $DSI$  is equal to one, whereas two volumes non-overlap volumes lead to the  $DSI$  of zero.

For a  $n$ -dimensional deformation fields  $D_n(x_1, \dots, x_n)$ , the Jacobian matrix  $J$  can be expressed as

$$J = \begin{bmatrix} \frac{\partial D_1}{\partial x_1} & \dots & \frac{\partial D_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial D_n}{\partial x_1} & \dots & \frac{\partial D_n}{\partial x_n} \end{bmatrix}$$

where  $x_1, \dots, x_n$  are the directions of the deformation vectors, for example, for 3D deformation fields,  $x_1, x_2, x_3$  correspond to the  $X, Y,$  and  $Z$  direction in Cartesian coordination system. The absolute value of the Jacobian determinant of the deformation field at a given point provides the information of volumetric change at the point. If the Jacobian determinant is less than one, the volume shrinks at the point. If the Jacobian determinant equals one, the volume at the point keeps constant. If the Jacobian determinant is large than one, the volume expands at the point. Jacobian determinant can also measure the orientation change in the deformation fields. The positive Jacobian determinant indicates the preservation of the orientation while the negative Jacobian determinant indicates the change of the orientation, such as folding. Tissue folding is

hardly observed during the normal organ deformation, thus, the Jacobian determinant provide a way to check whether the deformation is physically feasible.

3. Distance measures<sup>127, 132</sup>. The most straight forward distance measure is to compute the error between computed values and the gold standards. However, distance measures may not sufficient enough to characterize the deformation. Hellier et al.<sup>132</sup> reported using principal component analysis to characterize shape similarity.

### **3.1.4 Visual Verification**

Visual verification is the most common validation technique and should be used as routine ongoing validation approach at every opportunity<sup>134</sup>. Several approaches can be used to assist the visual verification, such as the difference image (the difference between the deformed source image and the target image), deformed mesh grid (applying deformation field to uniformly spaced grid to observe the deformation), quiver plot, and stream line (the deformation field is visualized as a stream of the fluid). The reported limits of visual verification were 2 mm in translation, 2 degree for in-plane rotation, and 4 degree for cross-plane rotation.

## **3.2 Robustness**

Robustness is defined as the performance change of a method in the presence of disruptive factors, such as intrinsic data variability, pathological change<sup>123</sup>, in other word, small variations of the input should result small variations of the output<sup>36</sup>. In DIR, robustness refers to the registration result should be insensitive to the noise level of the input images or the input parameters. The common input parameters of DIR are 1) image dataset; 2) registration parameters. Therefore, the robustness can be refereed to measure the variation of the result of the algorithm when the following conditions vary: 1) variation of the image quality, such as the image noise level; 2) variation of the registration parameters.

### 3.3 Consistency

Consistency studies the effect of transformation composition in the DIR. For instance, let  $T_{a,b}$  denote the transformation from image  $a$  to image  $b$ , the consistency is to study the composition of  $n$  transformations from the identity, that is, whether  $T_{1,2} \circ T_{2,3} \circ T_{3,4} \circ \dots \circ T_{n,1}$  is close to identity. If  $n$  is equals to 2, then consistency validation changes to  $T_{1,2} \circ T_{2,1}$ , which is the inverse consistency discussed in Chapter 2. More general group-wise consistency was studied by Geng et al. <sup>135, 136</sup>. Consistency measures do not require the existence of gold standard. However, it cannot be considered as the only validation approach in a validation study because significant difference deformation fields may result in the same consistency.

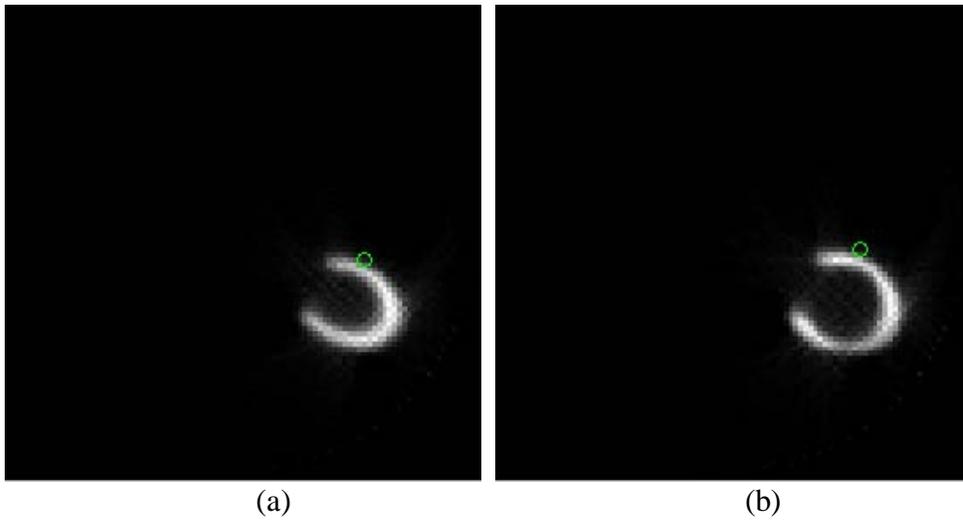


Figure 3-1. SPECT images of the dynamic physical phantom. (a) The frame 3 of the gated SPECT images with the location of one marker illustrated in green circle; (b) The location of the marker in (a) in the frame 8 of the gated SPECT images.

## CHAPTER 4

### A GENERALIZED INVERSE CONSISTENT CONSTRAINT FOR DEFORMABLE IMAGE REGISTRATION

In Chapter 2, we reviewed inverse consistent registrations, where most studies were focused on how to generate symmetric cost functions to reduce the inverse inconsistency. Here instead of explicitly compositing the symmetric cost function, we derived an inverse consistent constraint (ICC) using the fixed-points theory. Our new approach does not require the explicit computation of the inverse of the deformation field that might introduce numerical errors and result in long computation time. Moreover, the ICC does not involve any approximations for the inverse mapping and the robustness is not necessarily restricted by the parameter selection. Our method extends the previous work by Chen et al.<sup>107</sup> to the domain of inverse consistent registration. In Chen's work, an iterative approach based on the fixed-point theory was used to only compute the inverse of the deformation field. The rest of the chapter will be organized as the followings: the ICC will be first introduced, then the ICC methodology will be applied to two popular registration algorithms in medical imaging: the Demons<sup>30</sup> algorithm and the fast diffusion registration (FDR)<sup>32</sup> algorithm. A variation model (FDR-SIGMA) based on the likelihood estimation will also be introduced to improve the registration accuracy, together with its inverse consistent model (I-FDR-SIGMA). After that, the performances of the resulting inverse consistent demons (I-Demons), the inverse consistent fast diffusion registration (I-FDR), and the I-FDR-SIGMA algorithm will be evaluated with their respective unconstrained counterparts using a simulated phantom, a myocardium physical phantom, and clinical images. Finally, the comparison of Christensen's inverse consistent registration algorithm (CICR)<sup>66</sup> (discussed in Chapter 2.4) will be presented. To the best of our knowledge, the comparison with CICR has not been carried out by others authors on the topic of inverse consistency.

#### 4.1 The Inverse Consistent Constraint

In mathematics, a fixed-point of a function  $f$  refers to a function mapping to itself, i.e.,  $x$  is a fixed-point of  $f(x)$  if and only if  $f(x) = x$ . For example,  $x = 0$  is a fixed-point of the function  $f(x) = x^2 + x$ . To be an inverse consistent DIR (ICDIR), the transformation, which consists of the forward transformation  $h(x)$  followed by the backward transformation  $g(x)$ , is an identical transformation. Similarly, the backward transformation  $g(x)$  followed by the forward transformation  $h(x)$  is also an identical transformation. This inverse consistent relation can be expressed as

$$\begin{cases} g(h(x)) = x & x \in \Omega \\ h(g(y)) = y & y \in \Omega \\ h(x) = x + u(x) \\ g(y) = y + v(y) \end{cases} \quad (4.1)$$

where  $u(x)$  and  $v(x)$  are the displacement field of the forward transformation and backward transformation, respectively. Equation (4.1) indicates the location of any point in the source image should not change after the forward and the backward transformation, similar relationship applied to the point in the target image, where the location of a point should be invariable after the backward and the forward transformation. Equation (4.1) can be further simplified as

$$\begin{aligned} x &= g(h(x)) \\ &= g(x + u(x)) \\ &= x + u(x) + v(x + u(x)) \Rightarrow u(x) = -v(x + u(x)) \end{aligned} \quad (4.2)$$

Similar formulation can be derived for  $v(y)$

$$v(y) = -u(y + v(y)) \quad (4.3)$$

We call the Equation (4.2) and the Equation (4.3) the inverse consistent constraints (ICC), which must be satisfied when a DIR algorithm is inverse consistent. In order to incorporate the ICC into DIR algorithm, the following numeric scheme is proposed

1. Initialize the forward and the backward displacement field  $u^{i=0}$  and  $v^{i=0}$
2. Compute the forward displacement field  $u^{i+1}$

$$u^{i+1} = f(T(x), S(x), u^i) \quad (4.4)$$

3. Update the backward displacement field  $v^i$  by the forward field  $u^{i+1}$  using ICC

$$v^i = -u^{i+1}(x + v^i) \quad (4.5)$$

4. Compute the backward displacement field  $v^{i+1}$

$$\bar{v}^{i+1} = f^{-1}(S(x), T(x), v^i) \quad (4.6)$$

5. Update the forward displacement field  $u^{i+1}$  by the backward field  $v^{i+1}$  using ICC

$$u^{i+1} = -v^{i+1}(x + u^{i+1}) \quad (4.7)$$

6. Repeat step (2) through (5) until  $u$  and  $v$  converge.

where  $u^i$  and  $v^i$  are the forward displacement field and the backward displacement field at iteration  $i$ , respectively.  $S(x)$  is the source image,  $T(x)$  is the target image,  $f$  and  $f^{-1}$  are the iterative function to compute the forward and the backward transformation. In the next section, the ICC will be applied to construct the I-Demons, I-FDR, and I-FDR-Sigma algorithms.

## 4.2 The Inverse Consistent Demons (I-Demons) Algorithm

The first ICDIR was the I-Demons algorithm modeled by incorporating the ICC and the Demons algorithm. In the following, the detailed description of Demons, bijectivity Demons (B-Demons), and I-Demons will be provided.

### 4.2.1 Demons Algorithm

The Demons algorithm was proposed by Thirion<sup>30</sup> based on an analogy with thermodynamic concepts. The idea was to position the image entities called demons to move the image pixels by the local characteristics of the images in a similar way Maxwell did for solving thermodynamics equations. In the case of free-form deformation where all pixels with non-zero

gradient were selected as the demons, the displacement field  $u(x)$  (the mapping between the target image  $T(x)$  and the source image  $S(x)$ ) was estimated by

$$\frac{\partial u(x)}{\partial t} = \frac{(S(x+u(x)) - T(x))\nabla T(x)}{\|\nabla T(x)\|^2 + (S(x+u(x)) - T(x))^2} \quad (4.8)$$

where  $S(x+u(x))$  is the deformed source image, and  $\nabla$  is the gradient operator. At each iteration, the computed displacement field  $u(x)$  was convoluted with a Gaussian smoothing kernel for deformation field regularization. The Demons algorithm was shown to outperform other methods in a careful evaluation of inter-subject brain registration<sup>137</sup> by the global and local measures. The global measures included average volume, overlap of grey and white matter tissues, consistency of the deformation field; the local measures included segmentation of cortical sulci, visualization of deformed sulci, and numerical evaluation.

#### 4.2.2 Bijectivity Demons (B-Demons) Algorithm

To improve the inverse consistency of the deformation fields, Thiron<sup>30</sup> presented the bijectivity Demons (B-Demons) by first computing the forward transformation and the backward transformation using the Demons algorithm, then the residual transformations were calculated using the forward and the backward transformations, after that, adjusting the forward transformation and the backward transformation by subtracting half of the residual transformation to the forward transformation and adding the rest half to the backward transformation. On average, the inverse consistency residue could be reduced to less than one pixel. However, no detailed analysis was carried out, such as change in similarity measure with the iterations and computation time.

#### 4.2.3 I-Demons Algorithm

Following the numerical scheme presented in the Chapter 4.1, the I-Demons algorithm can be modeled utilizing the ICC. The numeric scheme of the I-Demons can be expressed as:

1. Initialize the forward and the backward displacement field  $u^{i=0}$  and  $v^{i=0}$ .
2. Compute the forward displacement fields  $u^{i+1}$  by Equation (4.8).
3. Update the backward displacement field  $v^i$  by Equation (4.5).
4. Compute the backward displacement field  $v^{i+1}$  by switching  $T$  and  $S$  in Equation (4.8).
5. Update the forward displacement field  $u^{i+1}$  by Equation (4.7)
6. Repeat step (2) through (5) until  $u, v$  converges.

### 4.3 The Inverse Consistent Fast Diffusion Registration (I-FDR) Algorithm

The second ICDIR model was the I-FDR algorithm constructed using the ICC and the FDR algorithm proposed by Fischer and Modersitzki<sup>32</sup>. In the FDR algorithm, the problem of image registration was modeled as the problem of finding the optimal deformation field  $u$ , which minimizes the following energy equation  $E(u)$

$$E(u(x)) = D(u(x)) + \lambda R(u(x)) \quad (4.9)$$

$$D(u(x)) = \frac{1}{2} \int_{\Omega} (S(x + u(x)) - T(x))^2 dx \quad (4.10)$$

$$R(u(x)) = \frac{1}{2} \sum_{j=1}^d \|\nabla u_j(x)\|^2 \quad (4.11)$$

where  $D(u(x))$  is the distance measure of the deformed image  $S(x + u(x))$  and the target image  $T(x)$  using the sum of the squared distance (SSD). The choice of  $D(u(x))$  depends on the image modalities, for example,  $D(u(x))$  can be in the form of mutual information for the multi-modality registration,  $d$  is the dimension of the displacement field  $u(x)$ ,  $d = 2$  for 2D displacement fields and  $d = 3$  for 3D displacement fields.  $R(u(x))$  is the regularization term to avoid discontinuous or suboptimal deformation fields and it is designed to penalize the deformation oscillations and consequently leads to a smooth deformation field.  $\lambda$  is the weighting factor of the regularization term. Using calculus of variations<sup>138</sup>, Equation (4.9) can be simplified as

$$\lambda \nabla^2 u(x) - [S(x + u(x)) - T(x)] \nabla S(x + u(x)) = 0 \quad (4.12)$$

By introduced the steady state solution  $\partial u(x,t)/\partial t = 0$ , Equation (4.12) could be solved using time marching method

$$\frac{\partial u(x,t)}{\partial t} = \lambda \nabla^2 u(x,t) - [S(x+u(x,t)) - T(x)] \nabla S(x+u(x,t)) \quad (4.13)$$

Equation (4.13) can be solved by two numeric schemes: the explicit scheme and the semi-implicit scheme. In the explicit scheme, Equation (4.13) can be written as

$$\frac{\partial u(x,t_{k+1})}{\partial t} = \lambda \nabla^2 u(x,t_k) - [S(x+u(x,t_k)) - T(x)] \nabla S(x+u(x,t_k)) \quad (4.14)$$

where  $u(x, t_k)$  is the displacement field  $u$  at iteration  $k$ . Solving Equation (4.14) is straightforward. The limitation of the explicit scheme is that it is only stable for very small step sizes, which leads to poor efficiency<sup>139</sup>. In the semi-implicit scheme, the limitation of the step size is reduced and the convergence is improved. The semi-implicit scheme can be written as

$$\frac{\partial u(x,t_{k+1})}{\partial t} = \lambda \nabla^2 u(x,t_{k+1}) - [S(x+u(x,t_k)) - T(x)] \nabla S(x+u(x,t_k)) \quad (4.15)$$

Equation (4.15) is indeed an inhomogeneous heat equation and can be effectively solved by additive operator splitting (AOS) scheme<sup>32</sup>, which achieves the accuracy of a conventional semi-implicit scheme while maintaining a linear complexity with respect to the image size. The benefit of AOS scheme is to decouple the non tri-diagonal equations into the summation of the tri-diagonal equations, which can be effectively solved by Thomas algorithm<sup>140</sup>. In the following, we will briefly describe how to use the AOS scheme to solve Equation (4.15).

### 4.3.1 AOS scheme

The AOS scheme was first introduced by Weickert<sup>141</sup> to improve the stability of the explicit scheme in the nonlinear diffusion filtering, which is only stable for small step sizes.

Equation (4.15) can be written as

$$\begin{aligned} \frac{V^{k+1} - V^k}{\tau} - \sum_{l=1}^d A_l V^{k+1} &= F_j^k \\ F_j^k &= -[S(x + u(x, t_k)) - T(x)] \nabla S(x + u(x, t_k)) \end{aligned} \quad (4.16)$$

where  $V^k = u_j(x, t_k)$  is the displacement vector for a fixed point  $j$ ,  $\sum_{l=1}^d A_l V^{k+1}$  is the numeric scheme of computing  $\lambda \nabla^2 u(x, t_{k+1})$  using finite difference. After rearranging Equation (4.16), we have

$$V^{k+1} = (I - \tau \sum_{l=1}^d A_l)^{-1} (V^k + \tau F_j^k), \quad k = 0, 1, \dots \quad (4.17)$$

The Equation (4.17) is a linear system with  $n$  unknown. Matrix  $A_l$  is tri-diagonal for each dimension, however the summation of  $A_l$  is not tri-diagonal matrix and Equation (4.17) cannot be directly solved by the efficient tri-diagonal solver: Thomas algorithm<sup>140</sup>. The idea of AOS is to replace the inverse of the summation of the matrix by the summation of the inverse. The AOS representation of Equation (4.17) can be expressed as

$$V^{k+1} = \frac{1}{d} \sum_{l=1}^d (I - d\tau A_l)^{-1} (V^k + \tau F_j^k), \quad k = 0, 1, \dots \quad (4.18)$$

Each component ( $l = 1, 2, \dots, d$ ) in Equation (4.18) can be efficiently solved by the Thomas algorithm.

To construct the I-FDR, similar update scheme described in Chapter 4.1 can be applied by replacing  $f$  and  $f^{-1}$  with Equation (4.15).

#### 4.4 A Robust Variational Model for Inverse Consistency

To achieve the optimal registration, the weighting factor  $\lambda$  of the energy function in Equation (4.9) is desirable to be adaptive during the iterations: smaller  $\lambda$  may produce unrealistic or discontinuous deformation field and larger  $\lambda$  may over smooth the deformation field and reduce the accuracy. Ideally it is desired to have large  $\lambda$  in the beginning of the registration when

the images are not well matched. With the progress of the iteration, the weighting factor  $\lambda$  needs to be gradually reduced as images are gradually aligned because large smoothness weighting factor may over smooth the deformation field and make it inaccurate. Moreover, both the Demons and FDR algorithms assume that the pixel intensity is preserved for the same object in image volumes. This assumption might not be true in reality, where pixel intensities may vary with the existence of image noise.

In order to make  $\lambda$  adaptive during the registration and account for some pixel intensity variations, we proposed a novel registration model (FDR-Sigma) incorporating a likelihood function  $L(u, \sigma)$  to estimate the similarity.  $L(u, \sigma)$  is defined as

$$L(u, \sigma) = \prod_{x \in \Omega} P(\{S(x + u(x)) - T(x), x \in \Omega\} | \sigma) = \prod_{x \in \Omega} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{|S(x+u(x))-T(x)|^2}{2\sigma^2}} \quad (4.19)$$

where  $\Omega \in R^n$  ( $n = 2$  or  $3$ ), the likelihood function  $L(u, \sigma)$  models the pixel intensities of the difference image,  $D(x) \triangleq S(h(x)) - T(x)$ , as independent and identically-distributed random variables from a Gaussian distribution with zero mean and standard deviation of  $\sigma$ ,  $\sigma$  is an parameter needed to be optimized. The log likelihood function  $L_{\log}(u, \sigma)$  can be written as

$$L_{\log}(u, \sigma) = \ln(L(u, \sigma)) = \int_{x \in \Omega} -\frac{|S(h(x)) - T(x)|^2}{2\sigma^2} dx - |\Omega| \ln(\sqrt{2\pi}\sigma) \quad (4.20)$$

Applying maximum likelihood estimation (MLE) to Equation (4.20) and incorporating the regularization term as shown in Equation (4.11), the proposed model is to find both the optimized  $u(x)$  and standard deviation  $\sigma$  to minimize the following energy function

$$E(u, \sigma) = \int_{x \in \Omega} \frac{\|S(x + u(x)) - T(x)\|^2}{2\sigma^2} dx + |\Omega| \ln(\sqrt{2\pi}\sigma) + \frac{\lambda}{2} \int_{x \in \Omega} \|\nabla u(x)\|^2 dx \quad (4.21)$$

Equation (4.21) can be solved using calculus of variations and time marching method,  $u(x)$  and  $\sigma$  can be computed iteratively using the following equations

$$\frac{\partial u}{\partial t} = \lambda \nabla^2 u(x) - \frac{(S(x + u(x)) - T(x)) \nabla S(x + u(x))}{2\sigma^2} \quad (4.22)$$

$$\sigma^2 = \frac{1}{|\Omega|} \int_{x \in \Omega} \|S(x + u(x)) - T(x)\|^2 dx \quad (4.23)$$

The numerical scheme can be expressed in the following: first,  $u(x)$  is initialized to zero and  $\sigma$  is computed using Equation (4.23), then, the updated  $\sigma$  is applied to update  $u(x)$  by Equation (4.22). Both  $u(x)$  and  $\sigma$  can be solved iteratively by applying Equation (4.22) and Equation (4.23). In the FDR-Sigma,  $\sigma$  is directly related to the difference image  $D(x)$ , i.e.,  $\sigma$  is small while two images are well aligned;  $\sigma$  is large while two images have large mismatch. This is a desirable characteristic because small  $\sigma$  leads to the reduced effect of the smoothing weighting factor  $\lambda$  for well aligned images and large  $\sigma$  leads to the increased effect of the smoothing weighting factor  $\lambda$  for misaligned images. Because of this adaptive mechanism, this MLE based approach could improve the algorithm convergence and reduce the sensitivity to the choice of weighting factor  $\lambda$ . The assumption of Gaussian distribution of the likelihood estimation makes it possible to allow some variations of pixel intensities in matching corresponding pixels.

Applying the ICC, the inverse consistent FDR-Sigma (I-FDR-Sigma) can then be constructed using methodology described in Chapter 4.1.

## 4.5 Algorithm Evaluations

### 4.5.1 Image Data for Evaluation

Three groups of image datasets were used for the algorithm evaluation:

1. Simulated phantom images

The simulated phantom images were synthesized using the mathematically simulated deformation described in Equation (3.1). The parameters of the simulated deformation field were  $m = 4$ ,  $b = 0.15$ . The source image was the representative 2D CT image of a prostate cancer patient. The source image (Figure 4-1 (a)) was cropped from the original CT image to exclude the patient couch and background, the size of the source image is 220 x 360 pixels. The target image (Figure 4-1 (b)) was generated by directly applying the simulated deformation field to the source image. The simulated deformation field is served as the gold standard for algorithm evaluation.

## 2. Myocardium phantom images

The images of myocardium phantom discussed in the Chapter 3.1.1 were used. There are totally 10 image frames in the myocardium image dataset, each image frame has a size of 96 x 96 x 34 pixels with a pixel dimension of 1 x 1 x 1 mm. Frame 3 and 8 are used as phantom images because of the large motion observed between two image frames.

## 3. Clinical 4DCT images

Ten 3D volumetric 4DCT images of a lung cancer patient were used in this study. 4DCT is capable of acquiring a sequence of ten phases of three-dimensional CT images over patient's breathing cycle. Phase 0 corresponds to the maximum inspiration, phase 50 corresponds to the maximum expiration, the duration of expiration is between phase 0 and phase 50, the duration of inspiration is between phase 50 and phase 0. All 10 phases are equally divided within the breathing cycle. In this study, phase 0 was the target image and the rest phase images (images from phase 10 to phase 90) were the source images. A region of interest (ROI) of 256 x 400 x 51 pixels was selected to exclude the background and the patient couch. The image pixel intensities were normalized between 0 and 1 before image registration.

### 4.5.2 Evaluation Criteria

The following criteria were selected to evaluate the effectiveness of the ICDIR:

#### 1. Average phantom error (APE)

APE measures the average errors between the computed marker motion and the true marker motion. It can be expressed as

$$APE = \frac{1}{M} \sum_{i=1}^M \|u_c(i) - u_t(i)\| \quad (4.24)$$

where  $u_c$  is the computed displacement,  $u_t$  is the true displacement, and  $M$  is the number of the markers. For the physical phantom images,  $M = 10$ ; for the simulated phantom images,  $M = 220 \times 360 = 79200$ , which is the number of pixels in the simulated images.

#### 2. Inverse consistency error (ICE)

The ICE is computed by

$$ICE = \frac{1}{2N} \sum_{i=1}^N (\|h \circ g(x_i) - id\| + \|g \circ h(x_i) - id\|) \quad (4.25)$$

where  $h(x)$  and  $g(x)$  are the forward and the backward transformation, respectively,  $N$  is the total number of the pixels within the volume,  $id$  is the identity matrix. Theoretically, ICE is expected to be zero if the registration is inverse consistent. However, ICE is difficult to be zero in reality, a small ICE is always preferred.

#### 3. Sum of squared error (SSE)

SSE measures the similarity between the deformed source image ( $S(x + u(x))$ ) and the target image ( $T(x)$ )

$$SSE = \int \|S(x + u(x)) - T(x)\|^2 dx \quad (4.26)$$

For a perfect match, SSE is zero. SSE is a global measure and may not be sensitive to local deformations.

#### 4. Robustness

Robustness is defined as the variation of the evaluation metrics, such as ICE or SSE, with respect to the change of the parameters, such as step size. A robust registration algorithm should be less sensitive to the parameter variation. Robustness and accuracy are two different concepts. A robust algorithm does not warrant its accuracy. Ideally a good algorithm is expected to be both accurate and robust. The standard deviation (SD) of the ICE, the APE, and the SSE with respect to the change of the step size will be evaluated. The evaluated algorithms are divided into two groups: 1) FDR-group algorithms (FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma) and 2) Demons-group algorithms (Demons, I-Demons, and B-Demons). The variation of step size is listed in Table 4-1 for the FDR-group and Demons-group algorithms using simulated phantom, physical myocardium phantom, and clinical images.

### 4.6 Results

#### 4.6.1 Simulated Phantom Images

The comparison of the FDR-group algorithms is shown in Figure 4-2. Figure 4-2 (a, c, e) shows the change in APE, ICE, and SSE at the 300<sup>th</sup> iteration with step size, respectively. Figure 4-2 (b, d, f) shows the change in APE, ICE, and SSE with the number of iterations at a step size corresponding to the minimum APE, respectively. When step size varied from 0.1 to 2.0, I-FDR-Sigma constantly outperformed the rest three algorithms (FDR-Sigma, FDR, and I-FDR) by the measure of APE and SSE. The accuracy (measured by APE) of the ICDIR (I-FDR and I-FDR-Sigma) was improved compared to their counterparts (FDR and FDR-Sigma). The minimum APE for I-FDR-Sigma was 7.48 pixels, which is 13% less than the minimum APE of FDR-Sigma (8.64 pixels), 25% less than the minimum APE of I-FDR (10 pixels), and 27% less than

the minimum APE of FDR (10.2 pixels). The minimum SSE for I-FDR-Sigma was 231, compared to 315 for FDR-Sigma, 3133 for I-FDR, and 3149 for FDR. The ICDIR had the reduced ICE compared to their counterparts. The minimum ICE for I-FDR-Sigma was 0.08 pixels, compared to 1.4 pixels for FDR-Sigma, 0.002 pixels for I-FDR, and 0.3 pixels for FDR. Figure 4-3 shows the plot of the energy for the FDR-group algorithms described by Equation (4.9) and Equation (4.21) using the optimal step size corresponding to the minimum APE, the energy for all four algorithms decreased with the iterations.

The robustness measurements of the FDR-group algorithms using the simulated phantom are shown in Figure 4-4. For APE, the maximum SD was 0.78 pixels (I-FDR-Sigma), the minimum SD was 0.11 pixels (FDR). For SSE, the maximum SD was 915 (I-FDR-Sigma), the minimum SD was 232 (FDR). For ICE, the maximum SD was 0.32 pixels (FDR-Sigma), the minimum SD was 0.0006 pixels (I-FDR). ICDIR (I-FDR and I-FDR-Sigma) significantly reduced the SD of ICE compared to their counterparts (Figure 4-4 (c)).

The comparison of the Demons-group algorithms using the simulated phantom is shown in Figure 4-5. After applying ICC, the I-Demons outperformed the Demons and B-Demons algorithms by APE, ICE, and SSE (Figure 4-5 (a, c, e)). I-Demons obtained the minimum APE of 4.6 pixels, which was 30% less than the minimum APE of B-Demons (6.6 pixels) and 32% less than the minimum APE of Demons (6.8 pixels). Using the optimal step size (step size = 1.0) corresponding to the minimum APE, at the 300<sup>th</sup> iteration, I-Demons had an ICE of 0.4 pixels, compared to 1.4 pixels for B-Demons and 2.9 pixels for Demons. Similarly, at the 300<sup>th</sup> iteration, SSE of I-Demons was 135, compared to 538 for Demons and 711 for B-Demons.

The robustness of the Demons-group algorithms using the simulated phantom is shown in Figure 4-6. I-Demons attained the minimum SDs of SSE (829) and ICE (0.17 pixels) among the

three Demons-group algorithms. However, the SD of APE for I-Demons (1.4 pixels) was larger than the SD of APE for Demons (1.0 pixels) and B-Demons (1.0 pixels).

#### **4.6.2 Physical Myocardium Phantom Images**

The comparison of the FDR-group algorithms using the physical myocardium phantom is shown in Figure 4-7. Based on the minimum APE, the optimal step size was 0.5 for I-FDR-Sigma, 0.3 for FDR-Sigma, and 2.0 for both I-FDR and FDR algorithms. As shown in Figure 4-7 (b), with the optimal step size, I-FDR-Sigma obtained the minimum APE of 0.38mm at the 300<sup>th</sup> iteration among all four algorithms, compared to 0.49 mm for FDR-Sigma, 0.44 mm for I-FDR, and 0.50 mm for FDR. As shown in Figure 4-7 (f), the SSE was 180 for I-FDR-Sigma and 181 for FDR-Sigma at the 300<sup>th</sup> iteration, which was smaller than the SSE of FDR (209) and I-FDR (227). Most significant improvement using ICC was observed by the measurement of the ICE. For both ICDIR (I-FDR and I-FDR-Sigma), ICE were greatly reduced from 0.017 mm for FDR-Sigma to 0.0008 mm for I-FDR-Sigma and from 0.015 mm for FDR to 0.0004 mm for I-FDR. Moreover, after 20 iterations, the ICE of both I-FDR-Sigma and I-FDR were converged to 0.0008 mm and 0.0004 mm, respectively. In contrast, the ICE of FDR-Sigma and FDR were linearly increasing after 50 iterations.

The robustness measurements of the FDR-groups algorithm using the physical simulated phantom are shown in Figure 4-8. I-FDR-Sigma achieved the minimum SD of APE (0.04 mm) among four algorithms. Both FDR-Sigma and I-FDR-Sigma had smaller SD of SSE (43 for FDR-Sigma and 46 for I-FDR-Sigma) than the SD of SSE for I-FDR (166) and FDR (157). The minimum SD of ICE was 0.0001 mm and it was achieved by the I-FDR algorithm.

The comparison of the Demons-group algorithm using the physical myocardium phantom is shown in Figure 4-9. The optimal step size corresponding to the minimum APE is 0.2 for all three Demons-group algorithms. At the 300<sup>th</sup> iteration, with the optimal step size, both Demons

and I-Demons had the similar APE (0.4 mm), which was 11% less than APE of B-Demons (0.45mm). For ICE, the I-Demons algorithm reduced the ICE by 68 times, from 0.70 mm for Demons to 0.01 mm for I-Demons. The minimum ICE was observed by B-Demons (0.005 mm). All three algorithms led to similar levels of SSE at the 300<sup>th</sup> iteration: 221 for Demons, 231 for I-Demons, and 213 for B-Demons.

The robustness measurements of the Demons-group algorithms were shown in Figure 4-10. In Figure 4-10 (a, c), the I-Demons algorithm attained the minimum SD of APE of 0.08 mm and SD of ICE of 0.02 mm among all three algorithms. In Figure 4-10 (b), all three algorithms led to similar levels of SD of SSE, 43 for Demons, 47 for I-Demons, and 41 for B-Demons, when step size varied from 0.1 to 1.0.

### 4.6.3 Clinical Images

Figure 4-11 shows the comparison of the FDR-group algorithms using the clinical 4DCT images (phase 0 and phase 50) of the lung cancer patient. With the optimal step size corresponding to the minimum SSE, at the 200<sup>th</sup> iteration, FDR-Sigma and I-FDR-Sigma attained SSE of 2049 and 1996, which were 40% less than SSE of I-FDR (3416) and FDR (3533). For ICE measurement, the I-FDR and I-FDR-Sigma algorithms attained the minimum ICE of 0.0009 mm and 0.007 mm, respectively, compared to 0.16 mm for FDR-Sigma and 0.037 mm for FDR. The robustness measures shown in Figure 4-11 (e, f) indicated that the I-FDR and I-FDR-Sigma led to smaller SD of ICE ( 0.002 mm for I-FDR-Sigma and 0.0002 mm for I-FDR), compared to SD of ICE of FDR-Sigma (0.04 mm) and FDR (0.01 mm)

Figure 4-12 shows the comparison of the Demons-group algorithms using the clinical 4DCT image of the lung cancer patient. The change in SSE and ICE with the number of iteration were listed in Figure 4-12 (b, d) with the optimal step size corresponding to the minimum SSE. I-Demons algorithm performed better than Demons and B-Demons. It attained the minimum SSE

of 1449, compared to SSE of 1808 for Demons and SSE of 1589 for B-Demons. It also attained the minimum ICE of 0.11 mm, compared to 0.25 mm for B-Demons and 1.59 mm for Demons. Figure 4-12 (e, f) shows the robustness measurements. Compared to Demons (1.79 mm), the SD of ICE for I-Demons was reduced by 90%, attained 0.17 mm at the 300<sup>th</sup> iterations. The SD of SSE for I-Demons was 336, which was larger than Demons (304), but smaller than B-Demons (340). I-Demons also converged faster than B-Demons and Demons in terms of SSE and ICE.

The result of registration from phase 0 to phase 10 through phase 90 of the 4DCT images is shown in Figure 4-13. The I-Demons, I-FDR, and I-FDR-Sigma algorithms attained lower SSE at the 200<sup>th</sup> iteration than the Demons, FDR, and FDR-Sigma algorithms, respectively, especially when the deformable motion was large, for example, registration of phase 0 to phase 50. Moreover, the I-FDR-Sigma and FDR-Sigma algorithms, based on the likelihood estimation, converged faster than the FDR and I-FDR algorithms (Figure 4-13 (c)).

#### **4.6.3.1 Visual estimation**

The effect of the ICC could be also demonstrated in Figure 4-14. Figure 4-14 (a, b) show one axial slice from the target and source images. The artifacts, indicated by the point A in Figure 4-14 (c) with the red arrow, was shown in the deformed image (Figure 4-14 (c)) using the demons algorithm with the parameters of  $\sigma = 0.5$  and step size = 1.0. In contrast, those artifacts were not visible in the deformed image (Figure 4-14 (d)) using the I-Demons algorithm with the same parameters. After plotted the profile of the displacement vector on axial direction (Figure 4-14 (e)), the artifacts was due to the large axial displacement vectors for some voxels computed by the Demons algorithm in the source images, causing these voxels to move out of the source images. Compared to Demons, I-Demons limited the motion magnitude (Figure 4-14 (e)) and produced more physiological consistent deformation field. The reduction of ICE using I-

Demons was shown in Figure 4-15 by the gray scale images. At the same gray scale, the I-Demons algorithm demonstrated superior ICE reduction than the Demons algorithm.

#### **4.6.3.2 Effects on re-contouring**

In the clinical auto re-contouring application, the contours of the gross tumor volume (GTV) were automatically propagated from the planning CT to the phase CT by the transformation computed using Demons, I-Demons, and B-Demons respectively. The details of the re-contouring were discussed in Chapter 5.1. The registration parameters were the following: the step size = 0.5, Gaussian kernel size = 3 x 3 x 3 with a standard deviation of 0.65, and number of iterations = 100. Figure 4-16 (a-b) show an axial slice of the overlay of the contours generated from the deformation fields by the I-Demons (aqua), the Demons (green), and B-Demons (red) with the planning contours transferred from the corresponding slice in the planning CT. Because of the respiratory motion, the planning contour (i.e., the brown contour) cannot represent the GTV in the phase CT. By visual assessment, the auto generated contours by three demons-group algorithms is able to match the anatomical boundary. The maximum difference of 1 mm was observed by measuring the Euclidean distance of the contour points between Demons and B-Demons algorithm (Figure 4-16 (c)). Maximum difference of 1 mm was observed by measuring the Euclidean distance of the contour points between Demons and B-Demons (Figure 4-16 (d)).

#### **4.6.4 Comparison of FDR, I-FDR, and Christensen's Inverse Consistent Registration (CICR)**

As discussed in Chapter 2.4, Christensen et al.<sup>66</sup> proposed an inverse consistent registration (CICR) algorithm by embedding the inverse consistency into the energy function. Among the proposed I-FDR, I-FDR-Sigma, and I-Demons algorithm, the I-FDR algorithm is most similar to the CICR algorithm. The difference between the I-FDR and CICR algorithms is

that 1) the I-FDR algorithm does not incorporate the ICC into the energy function, instead it applied the ICC to constrain the registration. 2) I-FDR used the L-2 norm of the displacement vector to regulate the displacement field while CICR used the elastic regulation. 3) I-FDR does not require explicitly computing the inverse of the deformation field. The CICR was implemented and compared to the FDR and the I-FDR algorithm, the algorithm evaluation was based on the physical myocardium phantom discussed in the Chapter 4.5.1 and the clinical 4DCT images. To the best of our knowledge, no direct comparison was published between the CICR and other inverse consistent DIR. Because the parameter selections can affect the registration results, the presented comparison is not to draw any conclusion of whether one algorithm is superior to the other, but to estimate their performance under the most available optimal parameters for each algorithm: for CICR, the parameters were:  $\rho = 10^{-4}$ ,  $\sigma = 10^7$ ,  $\chi = 5 \times 10^6$ ,  $\alpha = 10$ ,  $\beta = 10$ ,  $\gamma = 0$ , step size =  $5.0 \times 10^{-3}$ . For both FDR and I-FDR, the parameters were  $\lambda = 0.08$  and step size = 5.0. The image intensity was normalized in the range of 0 and 1 before DIR. All three algorithms were executed for 300 iterations.

The result of the physical myocardium phantom was shown in Figure 4-17 with the measures of SSE, APE, ICE, and computation time per iteration. Among them, the I-FDR algorithm achieved the best performance by the measurement of APE (Figure 4-17 (a)). The APE for the I-FDR, FDR, and CICR algorithms is 0.39 mm, 0.52 mm, and 0.61mm, respectively. The I-FDR algorithm achieved an APE reduction of 25% compared to the FDR algorithm and an APE reduction of 36% compared to the CICR algorithm. For ICE (Figure 4-17 (b)), the CICR, I-FDR, and FDR algorithm attained the ICE of  $6.9 \times 10^{-8}$ ,  $8.0 \times 10^{-4}$ , and  $2.2 \times 10^{-2}$ , respectively. Compared to the FDR algorithm where ICE increased with respect to the number of iterations, the ICEs of both CICR and I-FDR were converged after 20 iterations. Figure 4-17 (c) illustrated

the change in SSE with the number of iterations, all three algorithms achieved the similar level of SSE values after 300 iterations, 153.2 for FDR, 182.9 for CICR, and 180.0 for I-FDR. The computation time per iteration (TPI) was significantly different among three algorithms. CICR algorithm took the longest TPI (10.2 seconds), which was 5.4 times longer than I-FDR (1.6 seconds) and 19.4 times longer than FDR (0.5 seconds). The long TPI for CICR algorithm was due to the computation of the inverse deformation field and Fourier series calculation. Even though I-FDR did not achieve the minimum ICE and SSE, it had the advantage of achieving minimum APE, reducing the ICE by 27 times comparing with FDR and attained the reasonable computation time.

The comparison of three algorithms using 4DCT lung images is shown in Figure 4-18. Compared to SSE of  $1.23 \times 10^4$  for CICR, both I-FDR and FDR algorithm achieved similar SSE value,  $6.2 \times 10^3$  for I-FDR and  $6.5 \times 10^3$  for FDR. Compared to FDR, I-FDR reduced ICE by 97% (from 0.1275 pixels to  $3 \times 10^{-3}$  pixels) while CICR reduced ICE by 99% (from 0.1275 pixels to  $2 \times 10^{-4}$  pixels). However, CICR took the longest computation time (TPI = 185.2 seconds), compared to 25.9 seconds for I-FDR and 8.5 seconds for FDR. With carefully chosen parameters, CICR attained considerable good results; however, it is very challenge to optimize seven parameters in the CICR model. Moreover, its long computation time required the necessity of high performance computer<sup>142</sup> in order to reduce the computation time and meet memory requirements. In contrast, I-FDR is more computational efficient and achieves ICE of  $\leq 0.02$  mm in both clinical images and physical phantom evaluation.

#### **4.7 Discussion and Conclusion**

The proposed ICC provided a novel approach to constrain the inverse inconsistent registration to be inverse consistent and physiologically realistic. Compared to the Demons, FDR, and FDR-Sigma algorithms, the I-Demons, I-FDR, and I-FDR-Sigma algorithms

significantly reduced the ICE by up to 98%, 99%, and 96%, respectively. Small values of ICE indicate that the deformation is physiological consistent. In contrast, the ICE of Demons and FDR increased linearly with increasing iterations. Beside the ICE reduction, the ICC also improved the ICE robustness. The SD of the ICE of the ICDIRs was reduced by 2-50 times comparing with their counterpart, which indicated that the proposed ICC was successful in providing more physiological realistic deformations, reducing the sensitivity of the parameters selection. Even though the improvement of APE for ICDIR varied depending on the image dataset, the faster APE convergence of ICDIR was observed for both the simulated phantom and the physical myocardium phantom images. The I-FDR-Sigma algorithm provided the superior performance among the FDR-group algorithms, its likelihood estimation based similarity measures led to faster convergence. The fact that the I-Demons outperformed the B-Demons indicated that simple compensation of the inverse residue could not achieve the optimal deformation field. It was observed that the most of the ICE occurred on boundaries of objects, which might be due to the fact that the smoothness of the deformation field was difficult to preserve along the object boundaries because of the abrupt change of the gradient in the regions. The ICC can greatly improve the uniqueness of the correspondence between two images, and consequently reduce the ICE, which was consistent with the observation by Christensen et al.<sup>66</sup>. The superior performance of the ICC might be due to the fact that it provides a re-initialization mechanism to avoid the local minimum and achieves optimal convergence while traditional algorithm are more likely trapped in the local minimum. Provided a transformation is invertible, the new inverse consistent approach could lead to more consistent and physiologically meaningful deformation in medical imaging. It should be mentioned that the accuracy of the registration algorithms cannot be measured by ICE alone as ICE only measures the consistency

of the deformation fields and does not guarantee the similarity between the two images. For instance, zero deformation fields will always have an ICE of zero, however, zero deformation fields does not indicate the goodness of the registration. Therefore, ICE has to be combined with other similarity measures (such as SSE) in order to provide accurate evaluations of DIR algorithms.

The SSE value, as a global similarity measures, has been widely applied in published literatures to measure performance of registration algorithms. Practically, however, it is very difficult to choose the comparison criterion only based on the SSE value. For example, how does one choose the threshold of the SSE value difference to determine a better registration algorithm? Will a threshold value of 10 or 100 be enough? Will smaller SSE always indicates better registration? SSE, as shown in Equation (4.26), is an indirect measure of the goodness of the deformation fields and it may not be sensitive to the local change of the deformation fields. In Figure 4-19, we presented a case of comparing the registration results of two different DIR algorithms on a 2D slices from clinical lung 3D images. Method A is the Demons algorithm and method B is the FDR algorithm. At the end of 600 iterations, method A achieved a lower SSE value (90) than method B (668). Visual evaluation also indicated that the deformed image by method A (Figure 4-19 (c)) was more similar to the target image (Figure 4-19 (b)) than the deformed image by method B (Figure 4-19 (d)). However, the quiver plot of the deformation field (Figure 4-20) indicated that method B produced a smoother deformation while the deformation grid of method A looked rather irregular. Moreover, method A had the negative Jacobian determinant indicating the anatomical topology was not preserved in certain locations for method A. Therefore, only relying on SSE measurement itself might not reveal the characteristic of the deformation field. Even though method A had lower SSE value, method B

might be the better choice for the application where smoothness of the deformation field and preservation of topology are important.

The effects of the likelihood based variation model were also demonstrated using high noise myocardium phantom images. In Figure 4-21 (a) and (b), the FDR-Sigma and I-FDR-Sigma algorithms outperformed the FDR and I-FDR algorithms by the measurements of APE and SSE. A maximum APE reduction of 56 % was attained between I-FDR-Sigma and FDR, and a maximum SSE reduction of 53% was observed between I-FDR-Sigma and I-FDR. The ICE of all four algorithms were small: the maximum ICE among four algorithms was 0.0144 mm for FDR-Sigma, compared to 0.0069 mm for FDR,  $1.00 \times 10^{-4}$  mm for I-FDR, and  $4.78 \times 10^{-4}$  mm for I-FDR-Sigma. This again demonstrated that the likelihood function in the variation model allowed variations in pixels matching for high noise images, achieved fast convergence, and attained lower phantom errors.

We presented a novel ICC for DIR. We demonstrated that it did not require explicitly computing the inverse of the deformation fields, thus reduce the numeric errors. Compared to the unidirectional DIR, the ICC enabled DIR algorithms significantly reduced the inverse consistency error, and had less sensitive to parameter selections. The ICC demonstrated a new method to achieve physically realistic deformable registration. Although the current study was focused on the Demons, FDR, and I-FDR-Sigma algorithms, the new inverse consistency approach might be extended to the other deformable registration algorithms as well.

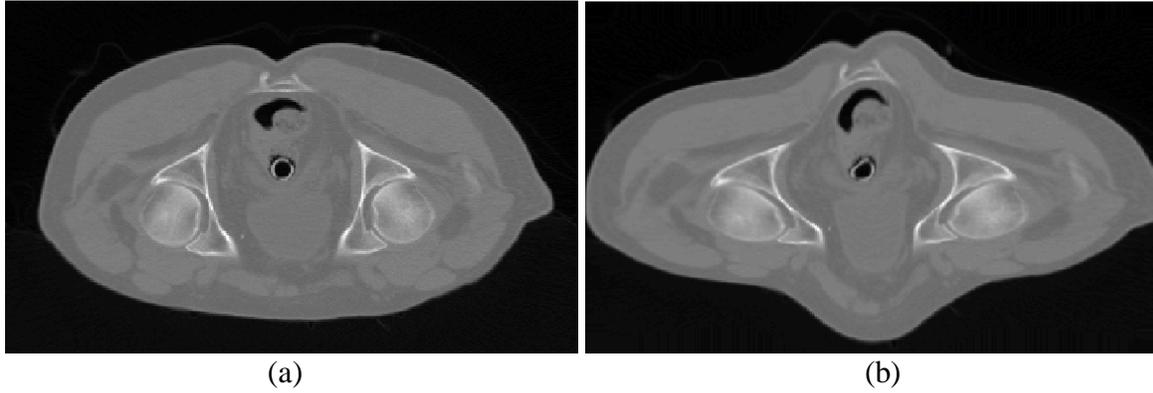


Figure 4-1. The simulated phantom. (a) The source image with an image size of 220 x 360 pixels. It is cropped from the representative 2D slice of the 3D CT images of a prostate cancer patient to exclude the patient couch and background. (b) The target image. The target image is generated by applying a mathematically generated deformation field to the source image (a). The target image has the same image size as the source image. The mathematically generated deformation field is considered as the gold standard for DIR evaluation.

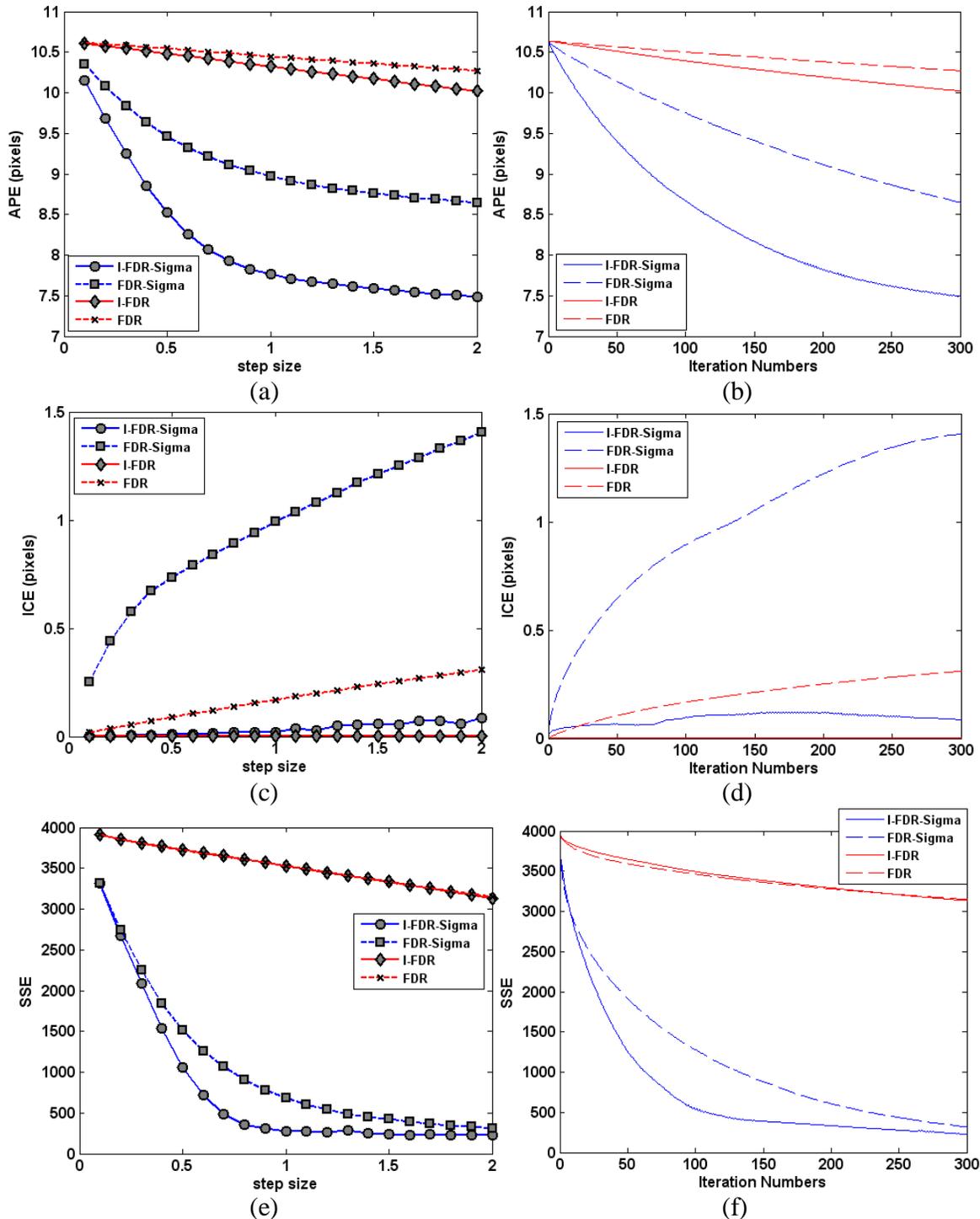


Figure 4-2. The evaluation of the FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma algorithms using the simulated phantom. (a) Change in APE at the 300<sup>th</sup> iteration with step size. (b) Change in APE with the number of iterations at a step size corresponding to the minimum APE. (c) Change in ICE at the 300<sup>th</sup> iteration with step size. (d) Change in ICE with the number of iterations at a step size corresponding to the minimum APE. (e) Change in SSE at the 300<sup>th</sup> iteration with step size. (f) Change in SSE with the number of iterations at a step size corresponding to the minimum APE.

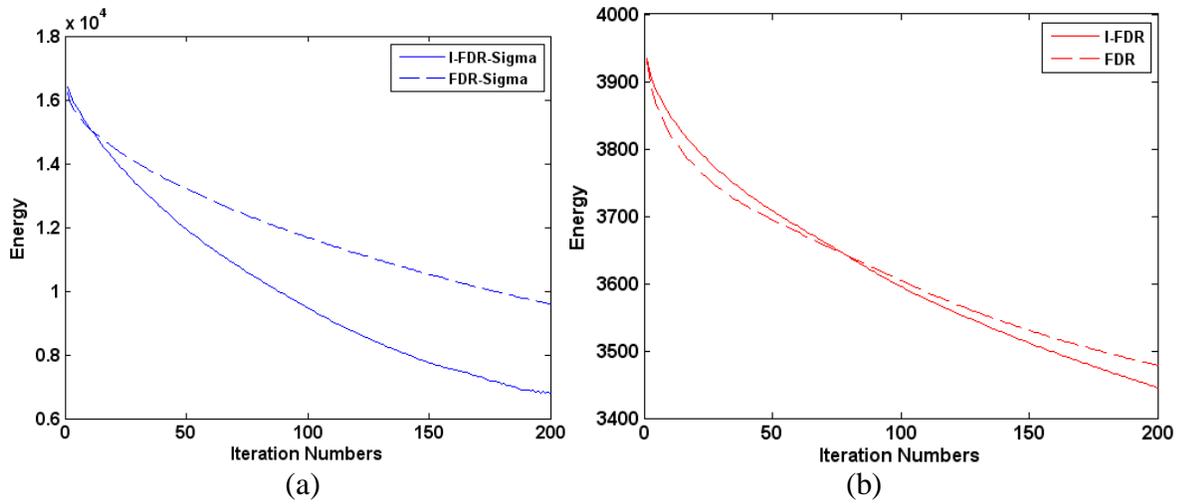


Figure 4-3. Change in energy with the number of iterations at a step size corresponding to the minimum APE. (a) Change in energy for the I-FDR-Sigma and FDR-Sigma algorithms. (b) Change in energy for the I-FDR and FDR algorithms.

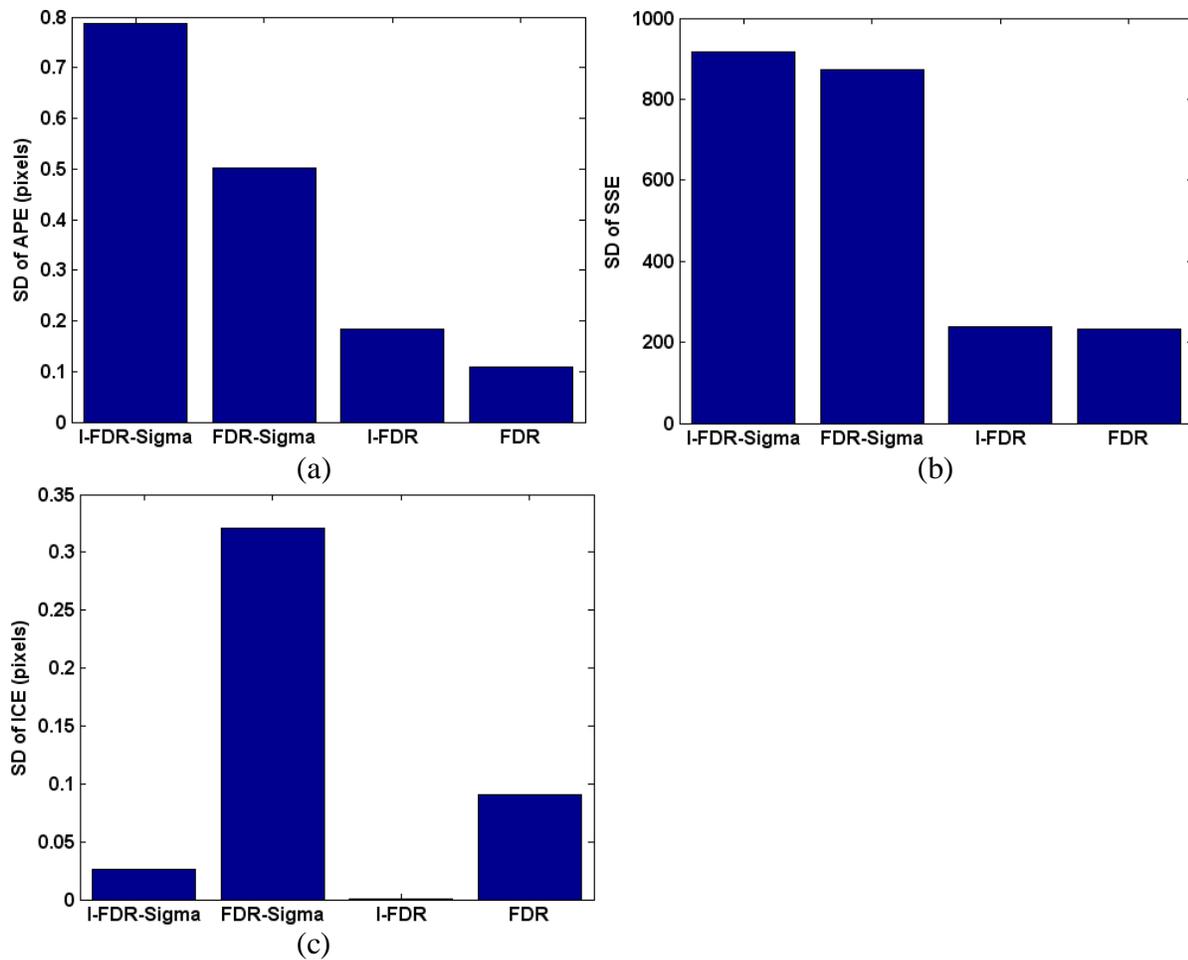


Figure 4-4. The standard deviations (SD) of the FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma algorithms with the change in step size using the simulated phantom. (a) SD of APE. (b) SD of SSE. (c) SD of ICE.

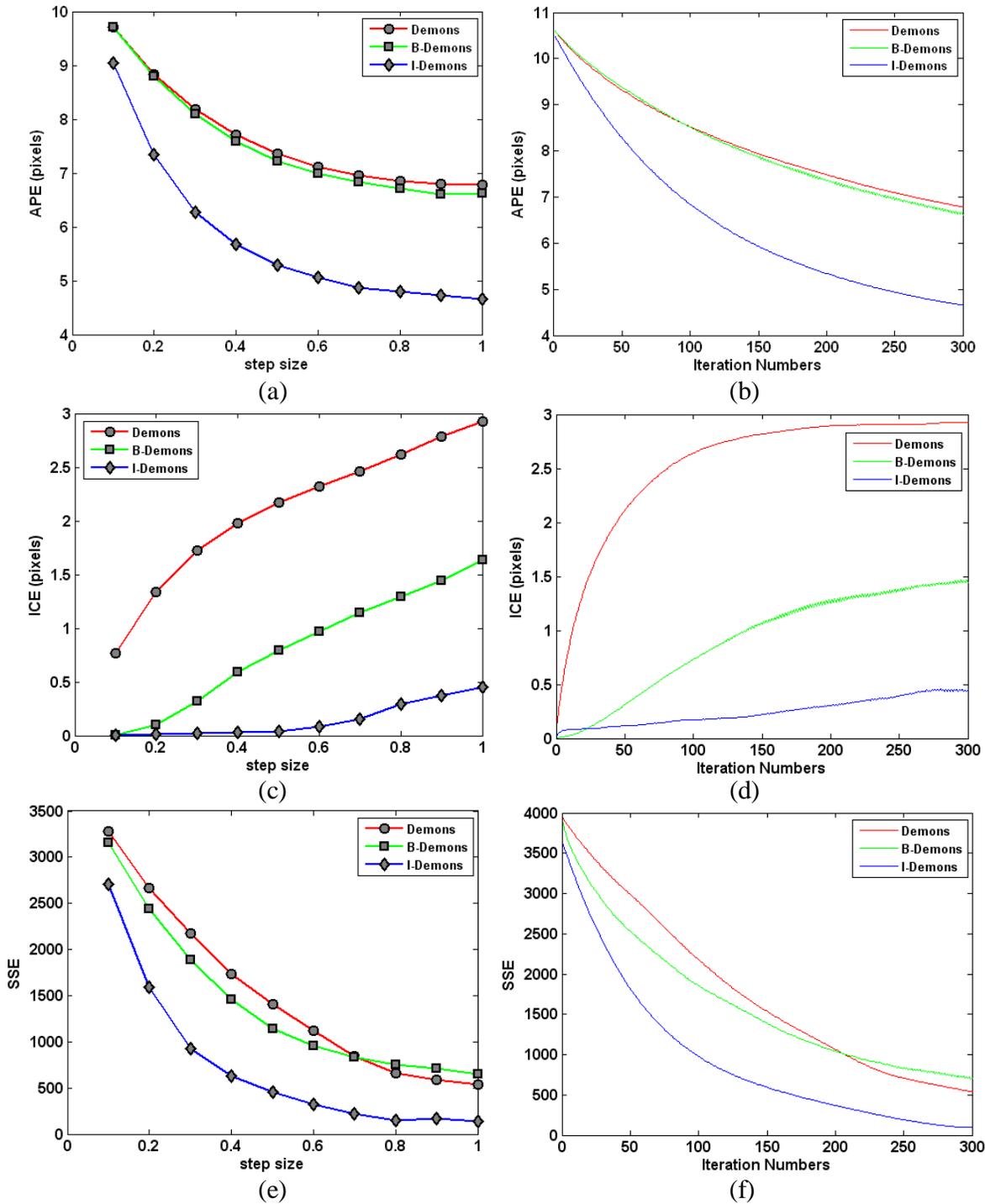


Figure 4-5. The evaluation of the Demons, I-Demons, and B-Demons algorithms using the simulated phantom. (a) Change in APE at the 300<sup>th</sup> iteration with step size. (b) Change in APE with the number of iterations at a step size corresponding to the minimum APE. (c) Change in ICE at the 300<sup>th</sup> iteration with step size. (d) Change in ICE with the number of iterations at a step size corresponding to the minimum APE. (e) Change in SSE at the 300<sup>th</sup> iteration with step size. (d) Change in SSE with the number of iterations at a step size corresponding to the minimum APE.

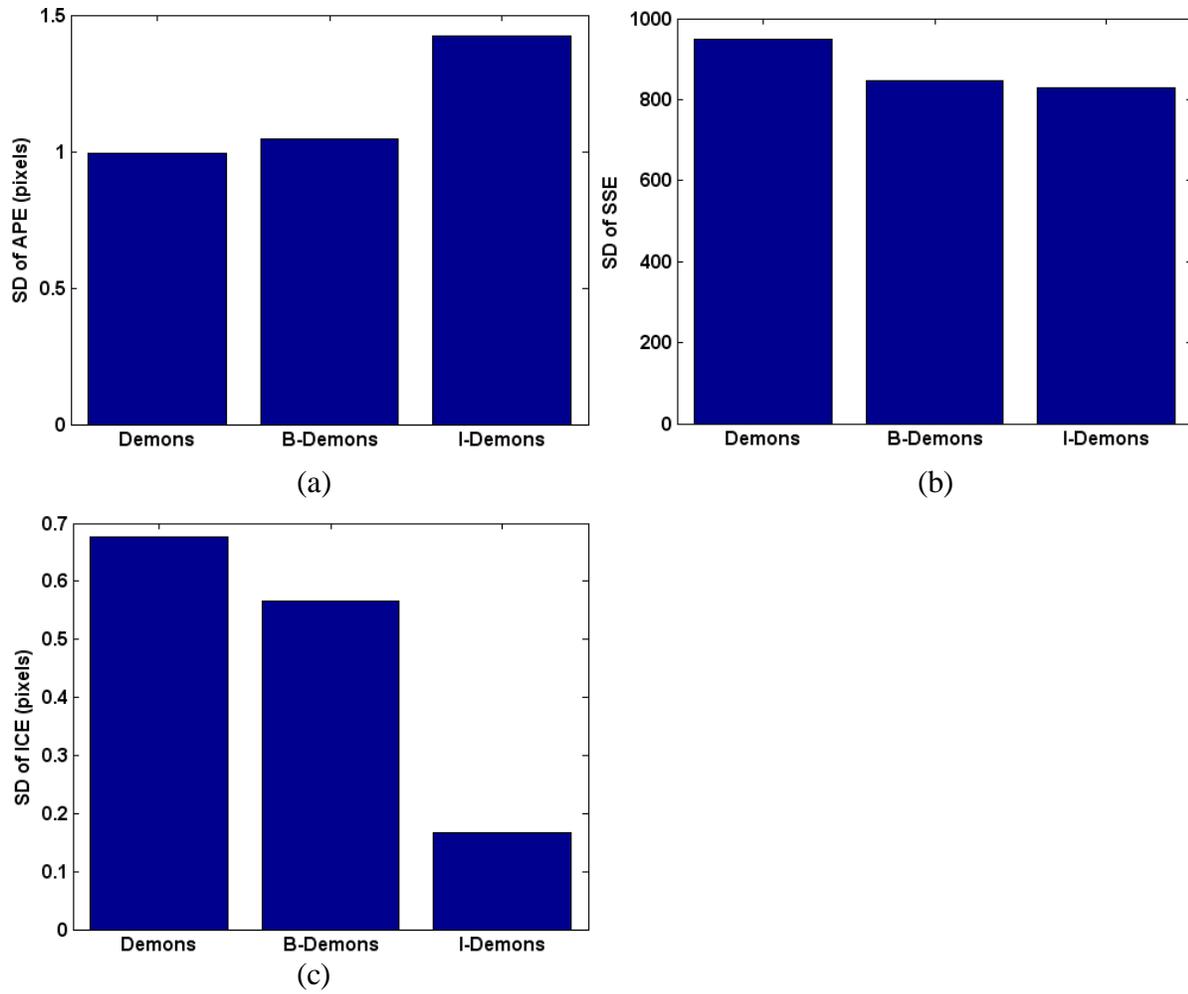


Figure 4-6. The standard deviations (SD) of the Demons, I-Demons, and B-Demons algorithms with the change in step size using the simulated phantom. (a) SD of APE. (b) SD of SSE. (c) SD of ICE.

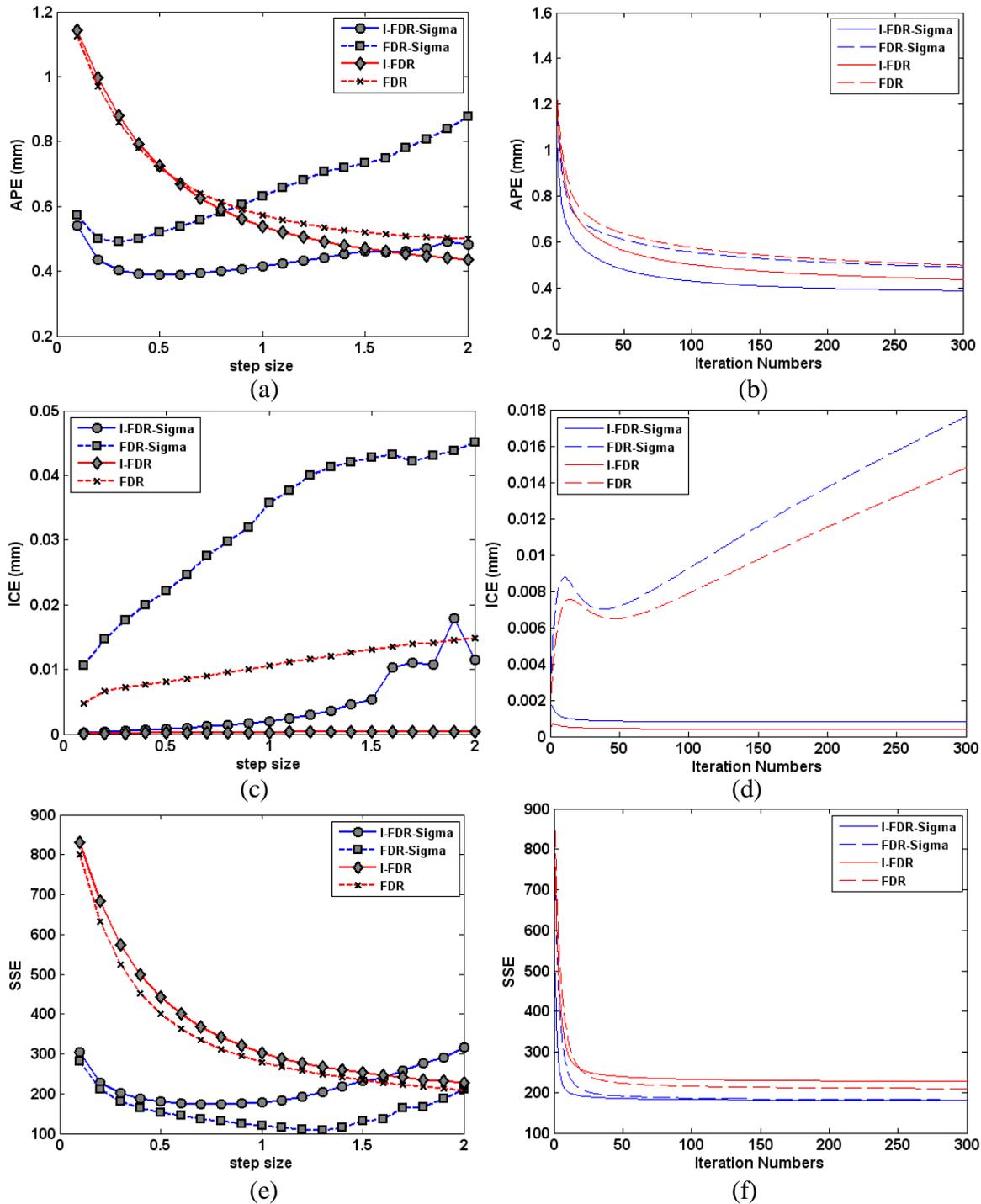


Figure 4-7. The evaluation of the FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma algorithms using the physical myocardium phantom. (a) Change in APE at the 300<sup>th</sup> iteration with step size. (b) Change in APE with the number of iterations at a step size corresponding to the minimum APE. (c) Change in ICE at the 300<sup>th</sup> iteration with step size. (d) Change in ICE with the number of iterations at a step size corresponding to the minimum APE. (e) Change in SSE at the 300<sup>th</sup> iteration with step size. (f) Change in SSE with the number of iterations at a step size corresponding to the minimum APE.

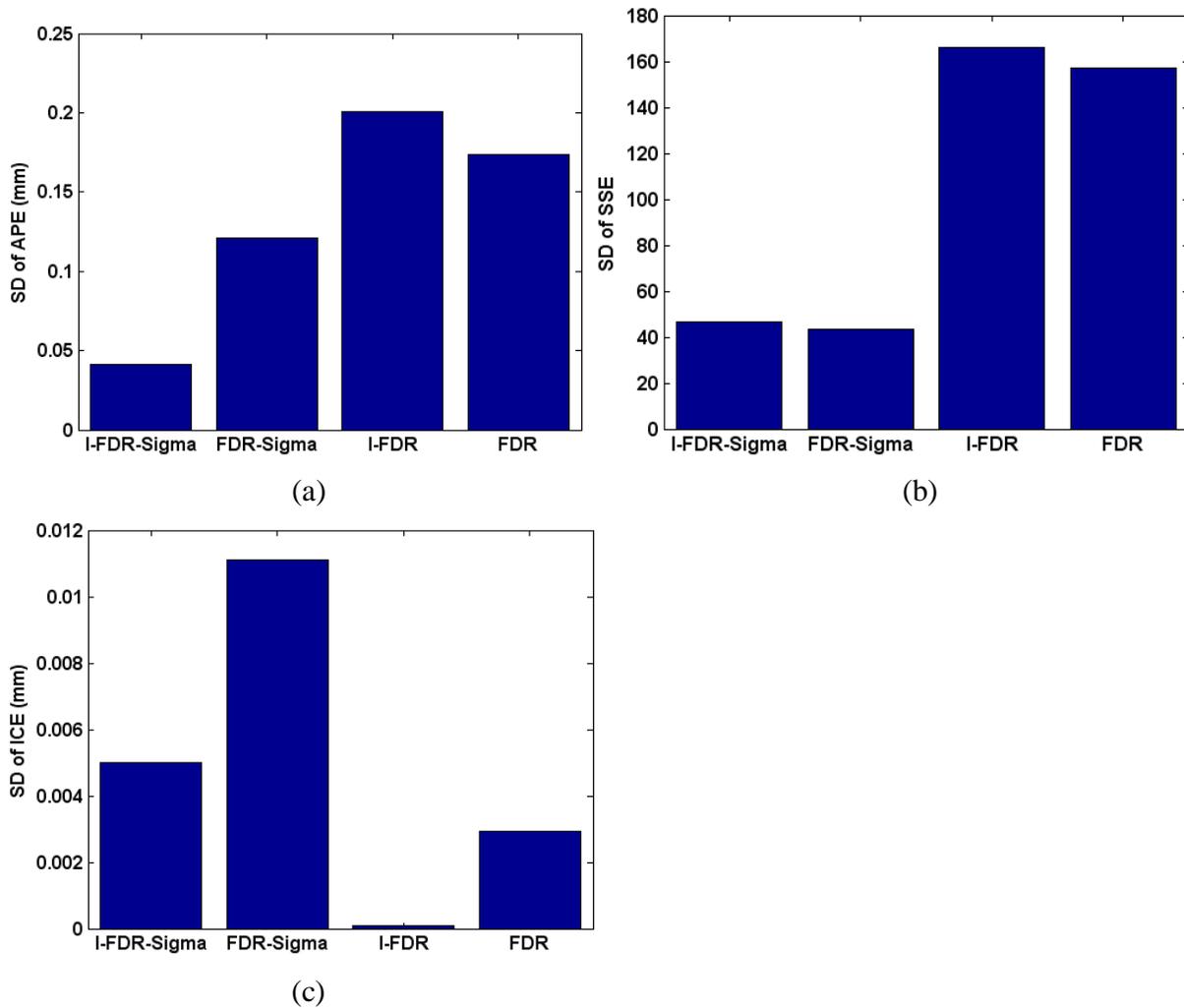


Figure 4-8. The standard deviations (SD) of the FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma algorithms with the change in step size using the physical myocardium phantom. (a) SD of APE. (b) SD of SSE. (c) SD of ICE.

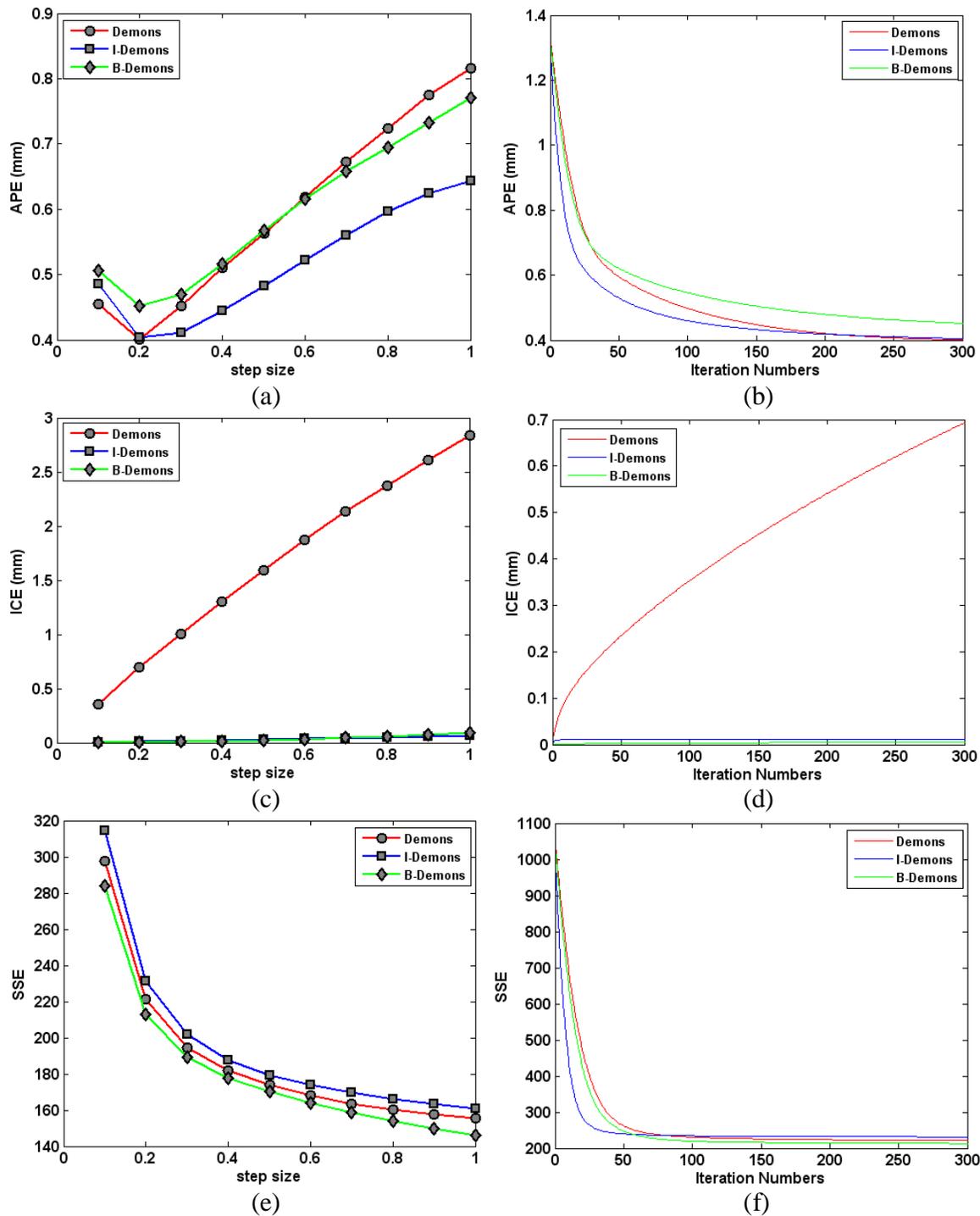


Figure 4-9. The evaluation of the Demons, I-Demons, and B-Demons algorithms using the physical myocardium phantom. (a) Change in APE at the 300<sup>th</sup> iteration with step size. (b) Change in APE with the number of iterations at a step size corresponding to the minimum APE. (c) Change in ICE at the 300<sup>th</sup> iteration with step size. (d) Change in ICE with the number of iterations at a step size corresponding to the minimum APE. (e) Change in SSE at the 300<sup>th</sup> iteration with step size. (f) Change in SSE with the number of iterations at a step size corresponding to the minimum APE.

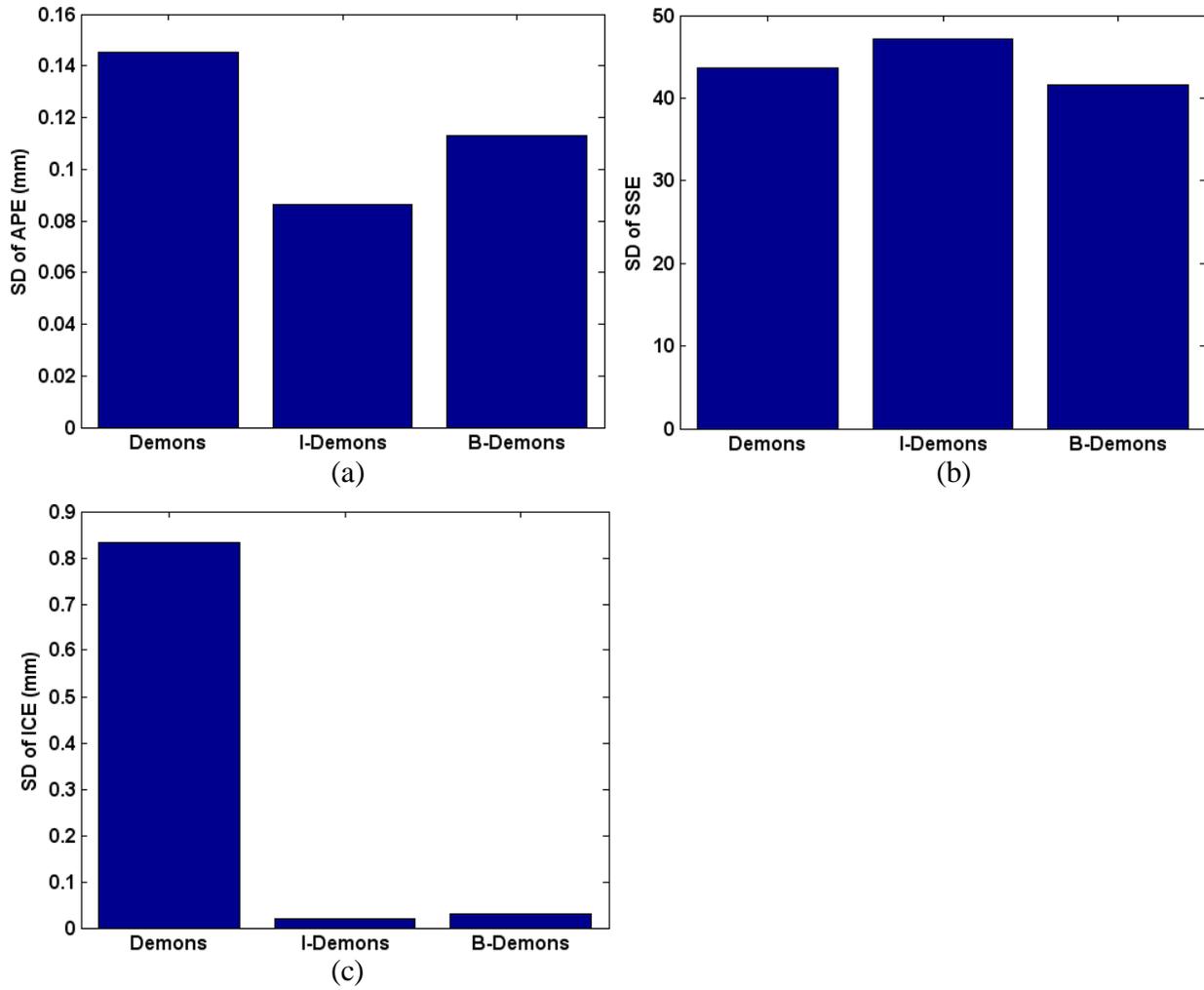


Figure 4-10. The standard deviations (SD) of the Demons, I-Demons, and B-Demons algorithms with the change in step size using the simulated phantom. (a) SD of APE. (b) SD of SSE. (c) SD of ICE.

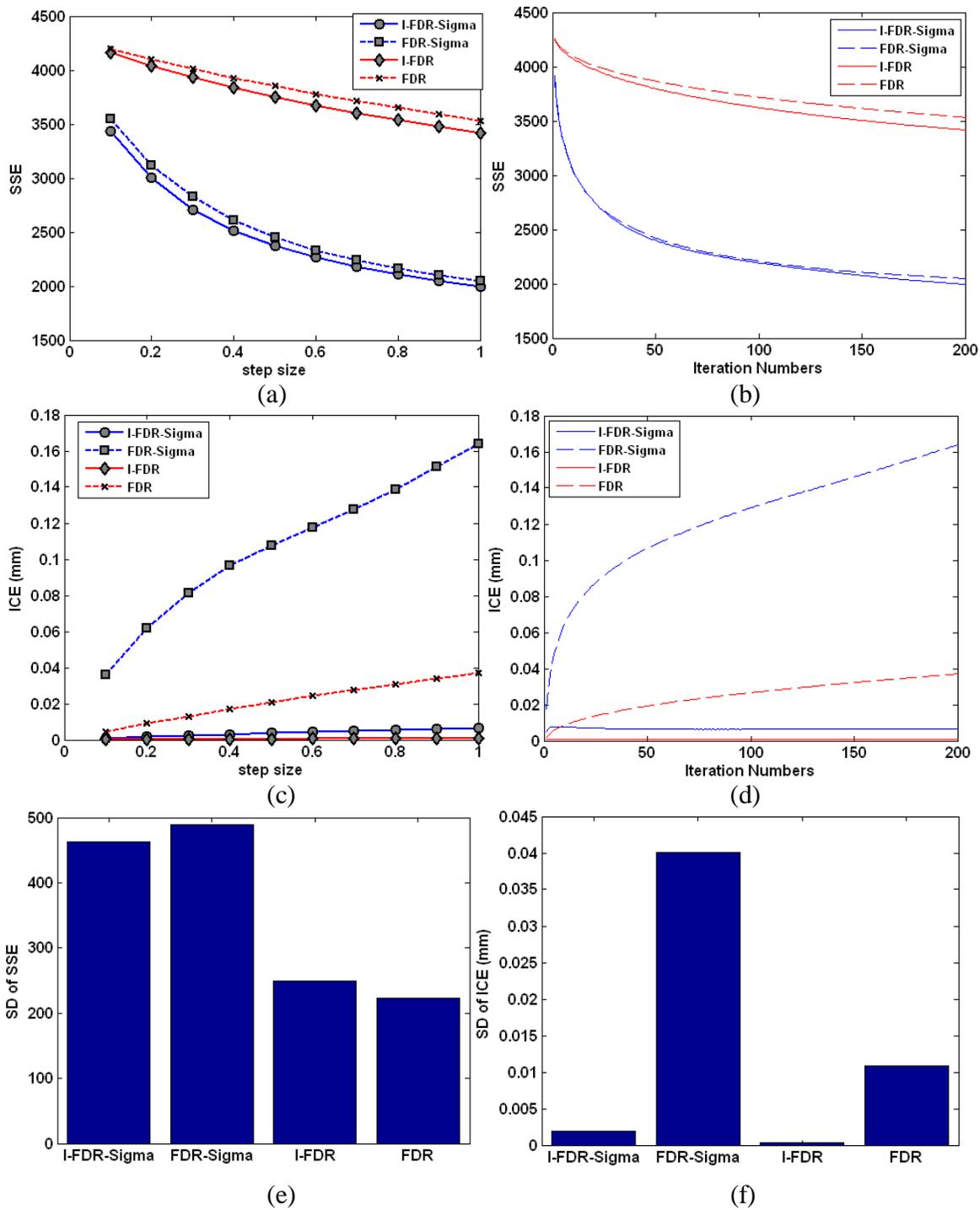


Figure 4-11. The evaluation of the FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma algorithms using the clinical 4DCT lung images. (a) Change in SSE at the 200<sup>th</sup> iteration with step size. (b) Change in SSE with the number of iterations at a step size corresponding to the minimum SSE. (c) Change in ICE at the 200<sup>th</sup> iteration with step size. (d) Change in ICE with the number of iterations at a step size corresponding to the minimum SSE. (e) SD of SSE with the change in step size. (f) SD of ICE with the change in step size.

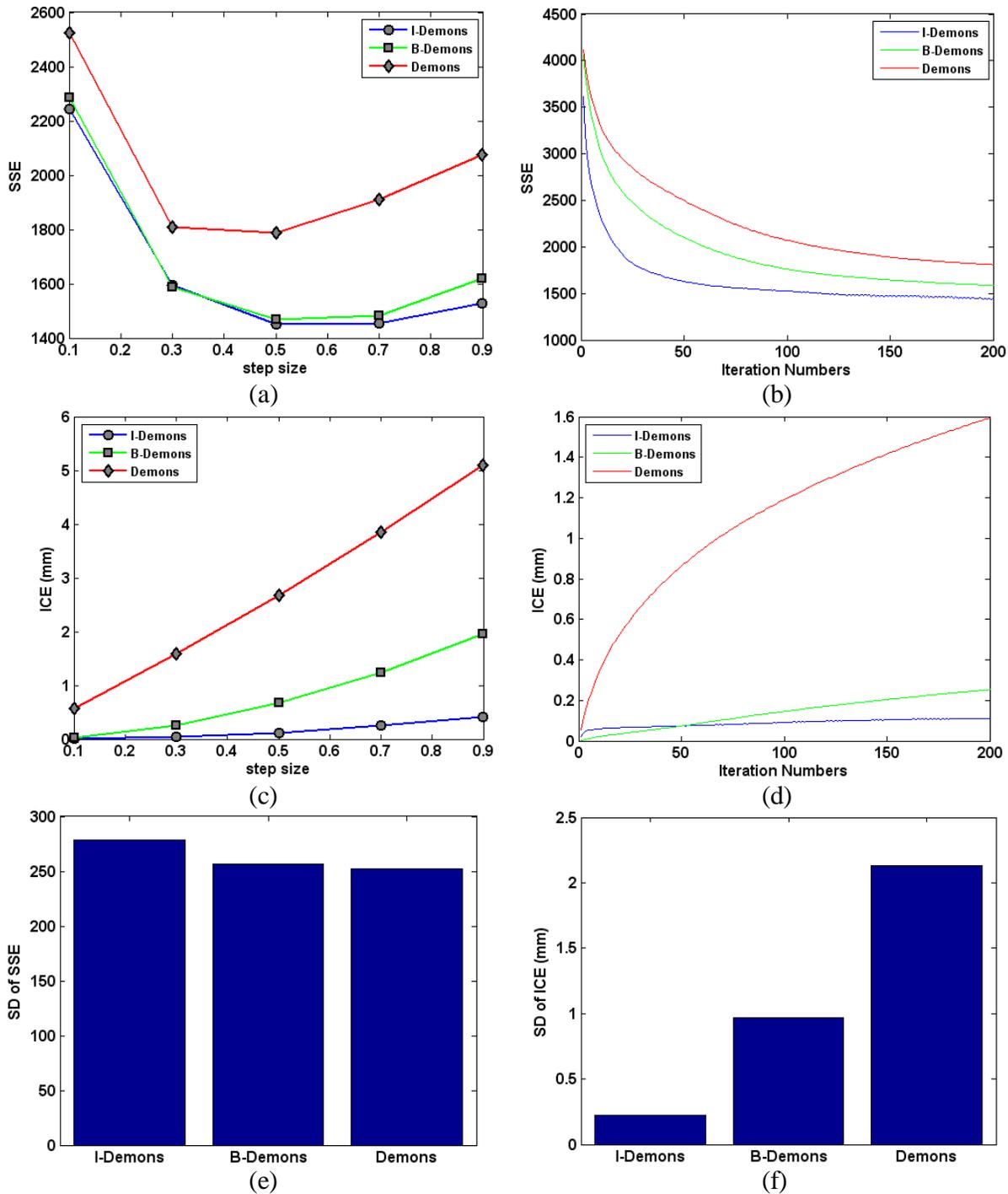


Figure 4-12. The evaluation of the Demons, I-Demons, and B-Demons algorithms using the clinical 4DCT lung images. (a) Change in SSE at the 200<sup>th</sup> iteration with step size. (b) Change in SSE with the number of iterations at a step size corresponding to the minimum SSE. (c) Change in ICE at the 200<sup>th</sup> iteration with step size. (d) Change in ICE with the number of iterations at a step size corresponding to the minimum SSE. (e) SD of SSE with the change in step size. (f) SD of ICE with the change in step size.

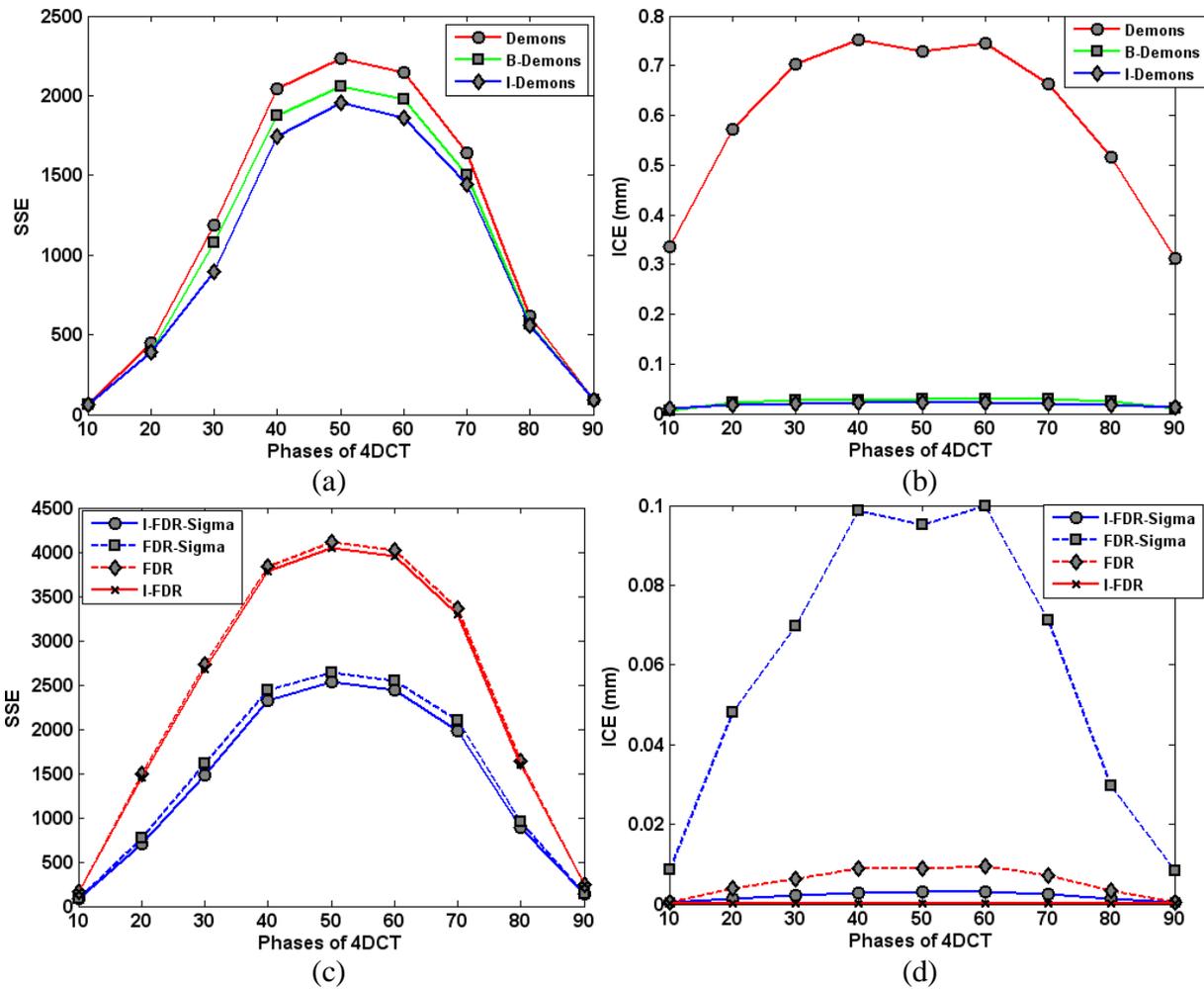


Figure 4-13. The evaluation of the Demons-group and FDR-group algorithms using 4DCT clinical lung images for 200 iterations. The 4DCT images contain 10 phases (phase 0, 10, ... , 90) corresponding to the patient's breathing cycle. Deformable registration was carried out between the phase 0 and the rest phases (phase 10 through phase 90). (a) Change in SSE of the Demons-group DIR algorithm with different phases of 4DCT at the 200<sup>th</sup> iterations. (b) Change in ICE of the Demons-group DIR algorithm with different phases of 4DCT at the 200<sup>th</sup> iterations. (c) Change in SSE of the FDR-group DIR algorithm with different phases of 4DCT at the 200<sup>th</sup> iterations. (d) Change in ICE of the FDR-group DIR algorithm with different phases of 4DCT at the 200<sup>th</sup> iterations.

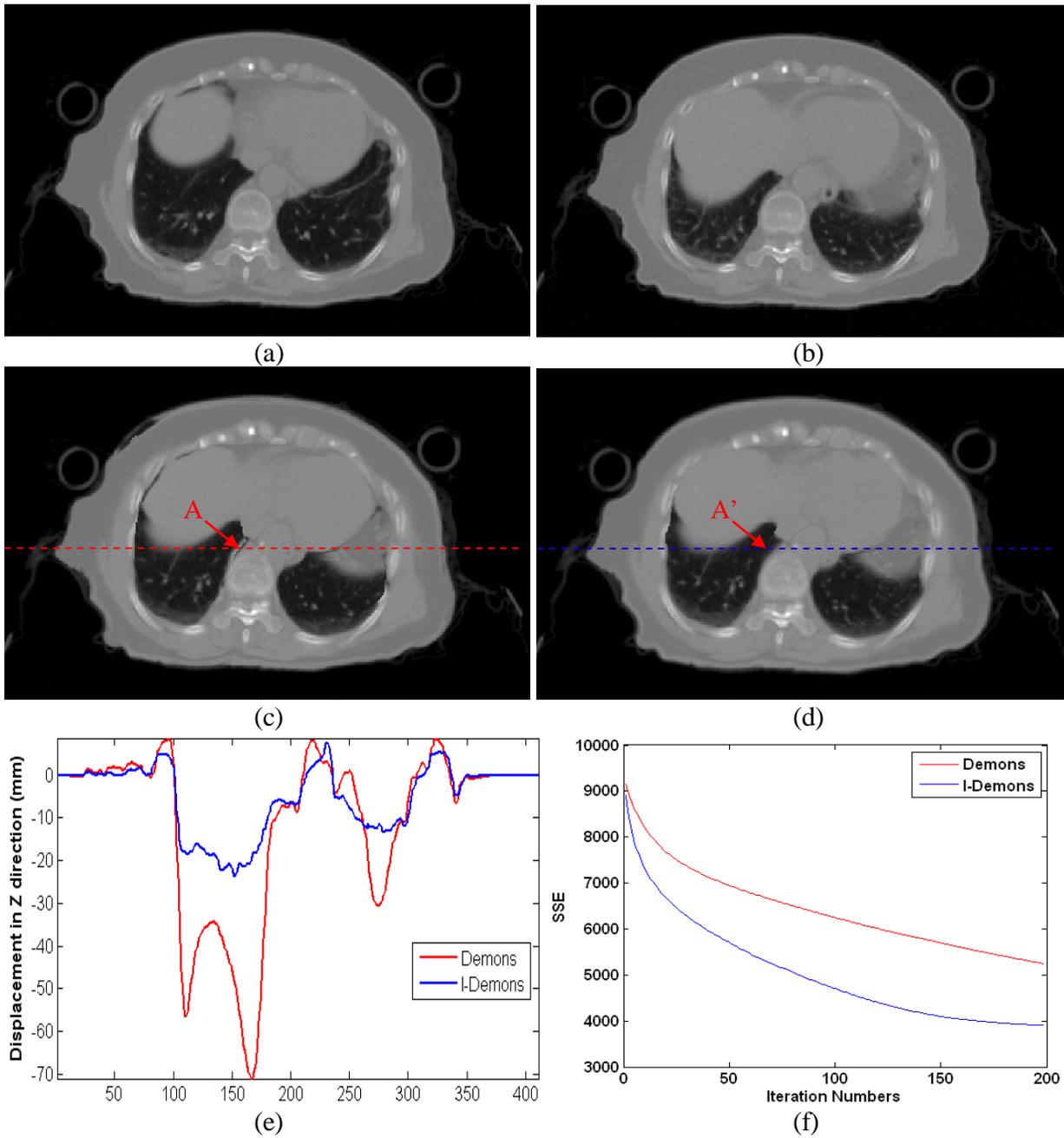


Figure 4-14. Visual comparison of DIRs using 4DCT dataset of a lung cancer patient. (a) Source image corresponding to the end-of-exhale phase; (b) Target image corresponding to the end-of-inhale phase; (c) Deformed image using Demons, the red arrow (point A) shows the artifacts; (d) Deformed image using I-Demons, the artifacts were not visible for I-Demons. (e) The profile of the displacement field on axial direction by Demons and I-Demons algorithms. (f) Change in SSE with the number of iterations.

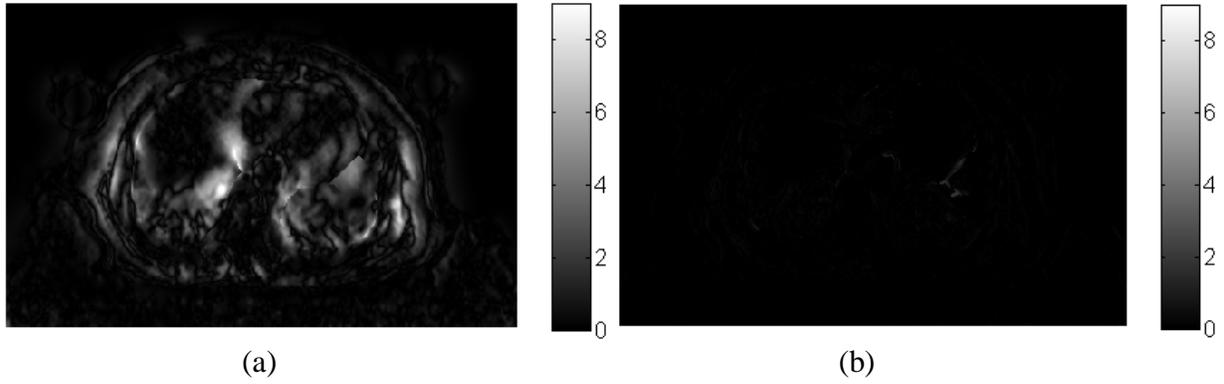


Figure 4-15. Distribution of the inverse consistency error (ICE) in gray scale. Higher intensity region indicated the larger ICE. (a) ICE by the Demons algorithm. (b) ICE by the I-Demons algorithm, The ICE was greatly reduced by the I-Demons comparing with the Demons.

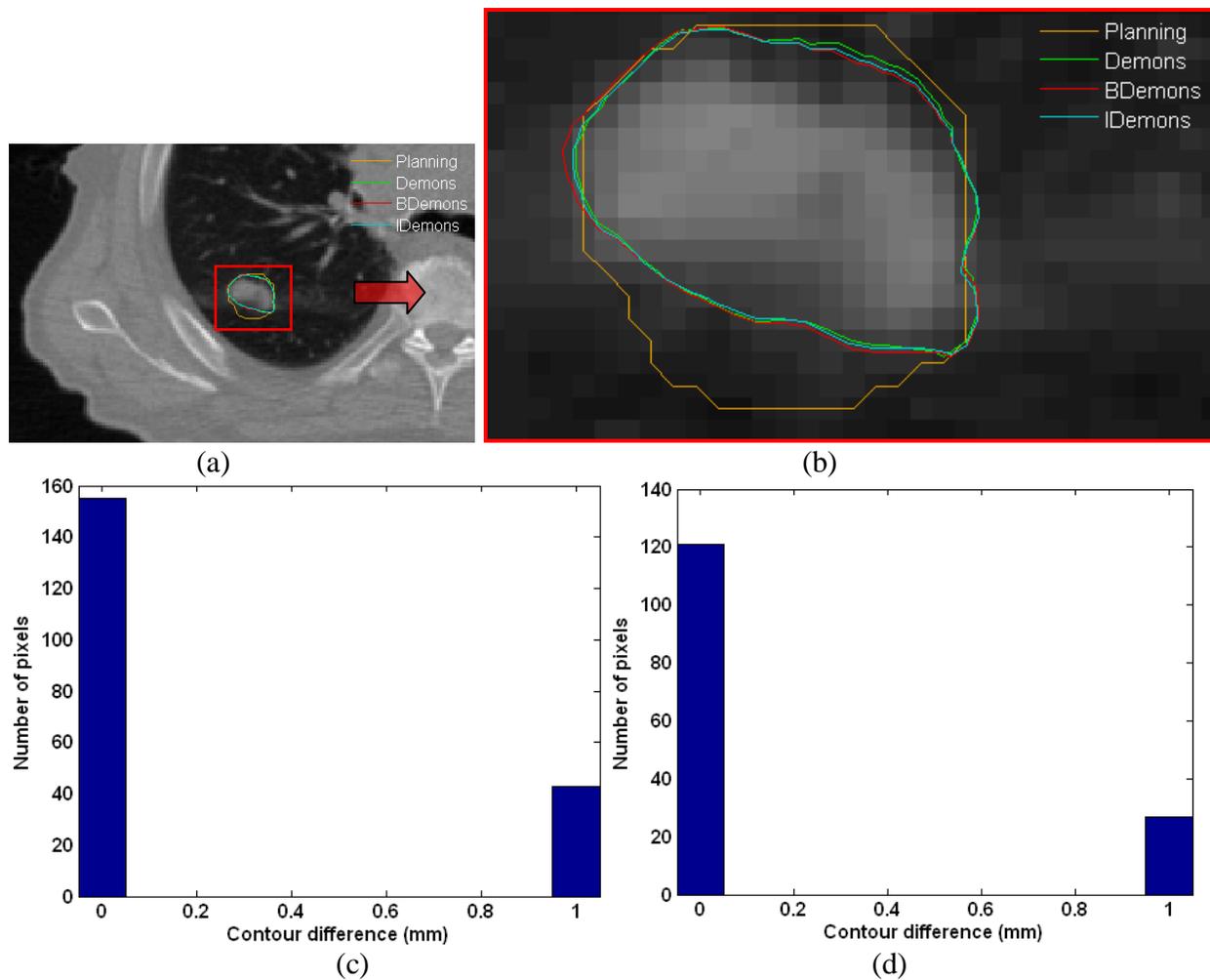


Figure 4-16. Auto re-contouring of gross tumor volume (GTV). The registration parameters were the following: the step size = 0.5, Gaussian kernel size = 3 x 3 x 3 with a standard deviation of 0.65, and number of iterations = 100. (a) The yellow contours represent the planning GTV, the green, aqua, and red contours represent the auto-generated GTV using Demons, I-Demons, and B-Demons, respectively. (b) The close-up of the red rectangle region in (a). (c) Histogram of the distance between the contour points generated by Demons and B-Demons. (d) Histogram of the distance between the contour points generated by Demons and I-Demons. The pixel dimension is 1 mm by 1 mm on X and Y directions.

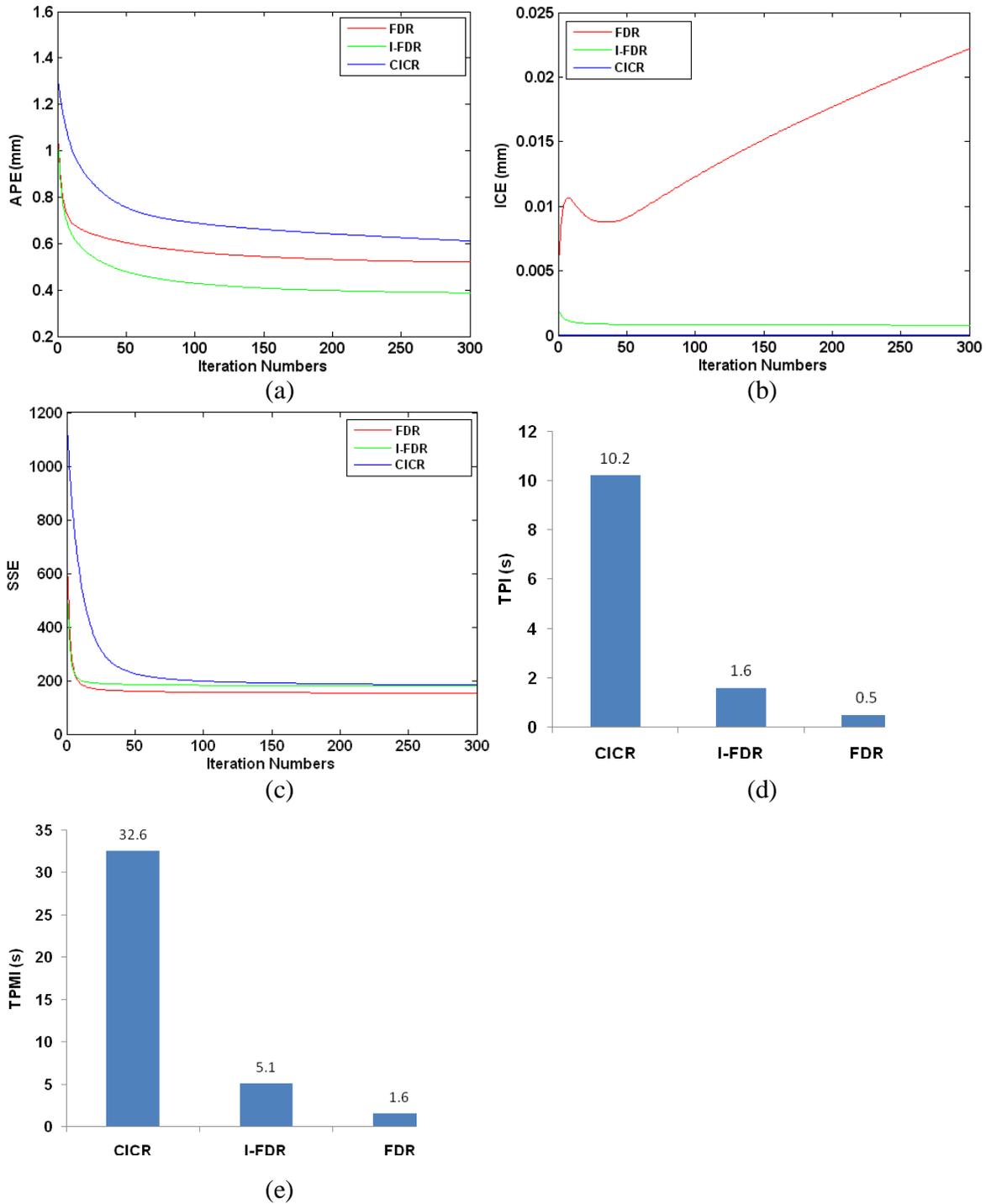


Figure 4-17. Comparison of the FDR, I-FDR, and CICR algorithms using the physical myocardium phantom. (a) Change in APE with the number of iterations. (b) Change in ICE with the number of iterations. (c) Change in SSE with the number of iterations. (d) Computation time per iteration (TPI) for three algorithms using an Intel 2.4GHz PC with 4GB memory running Windows XP operating system. (e) Time per million pixels per iteration (TPMP) for three algorithms.

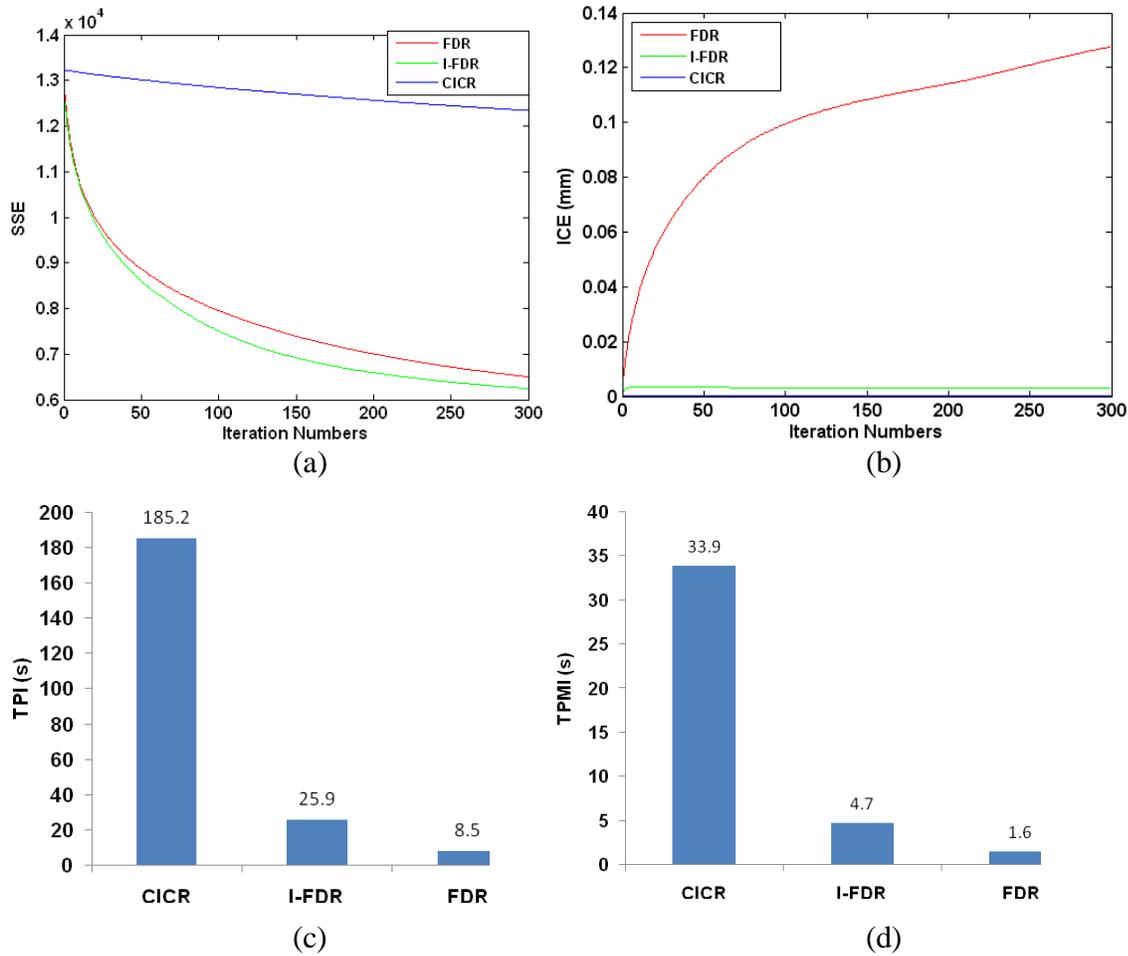


Figure 4-18. Comparison of the FDR, I-FDR, and CICR algorithms using 4DCT images from a lung cancer patient. (a) Change in SSE with the number of iterations. (b) Change in ICE with the number of iterations. (c) Computation time per iteration (TPI) for three algorithms using an Intel 2.4GHz PC with 4GB memory running Windows XP operating system. (d) Time per million pixels per iteration (TPMI) for three algorithms.

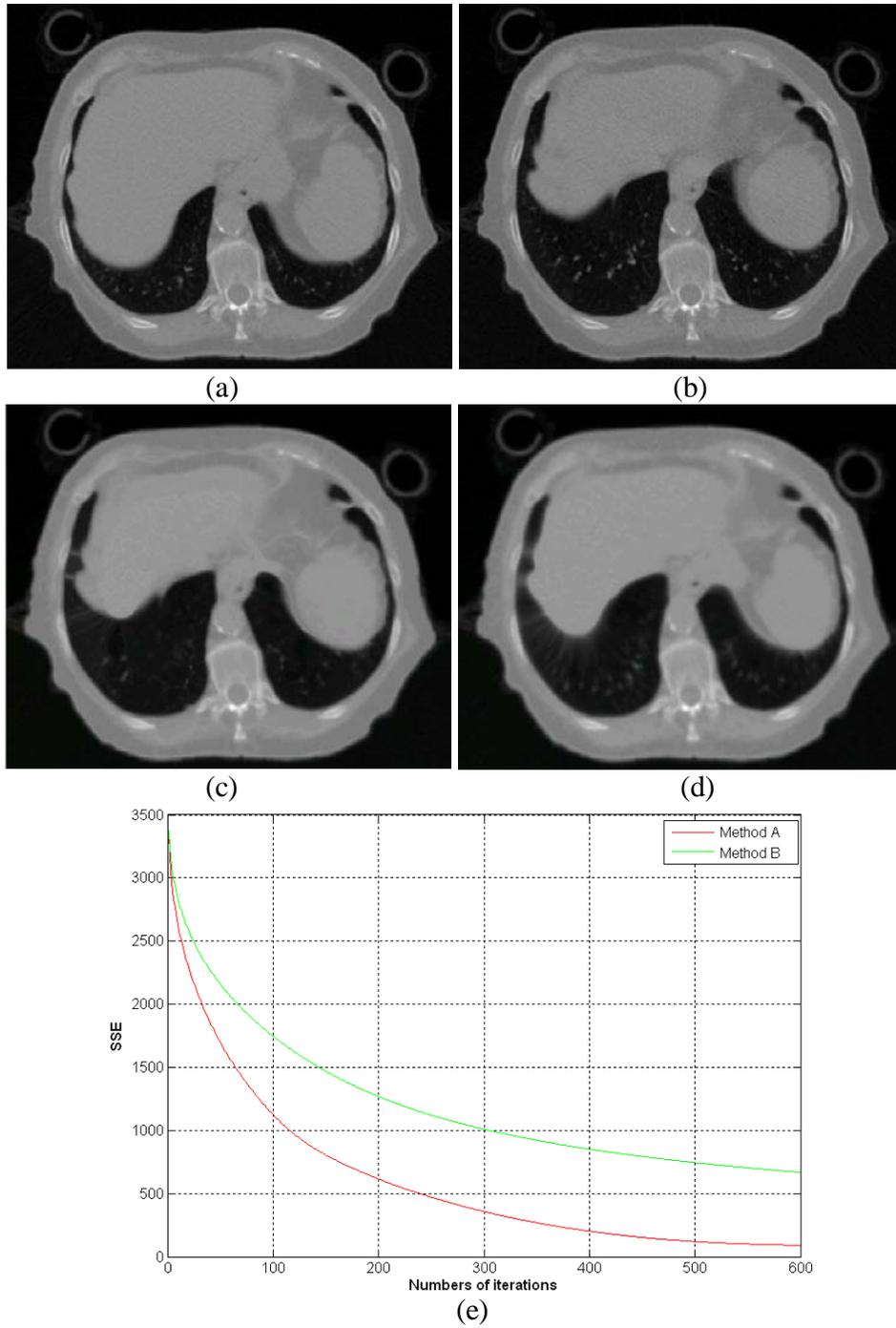
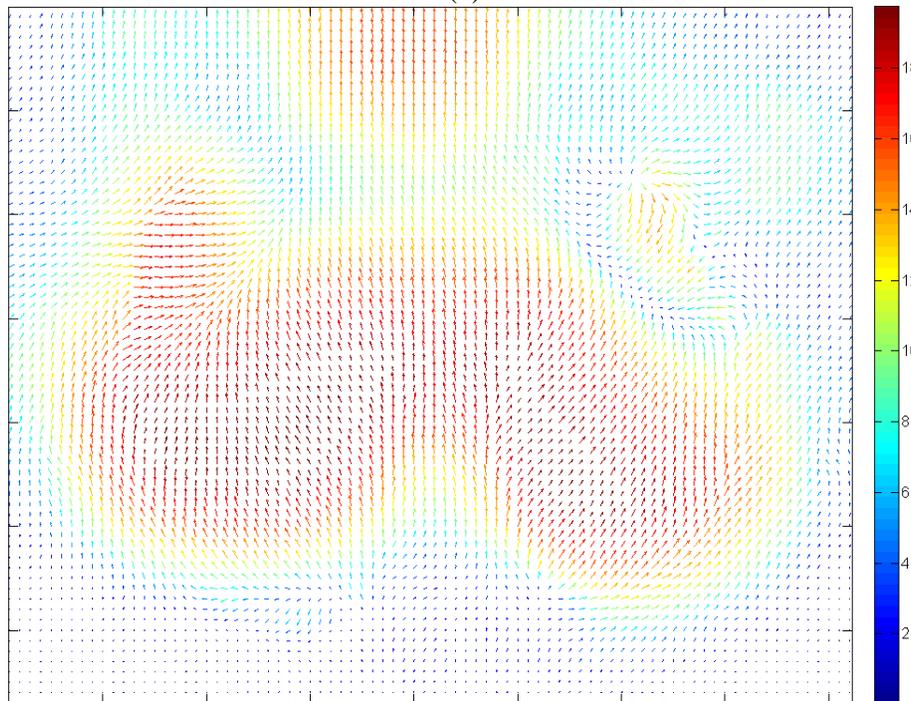


Figure 4-19. Comparison of two DIR algorithms using SSE and visual estimation. (a) Source image; (b) target image; (c) deformed source image using method A; (d) deformed source image using method B; (e) Change in SSE with the number of iterations for method A and method B.



(a)



(b)

Figure 4-20. Quiver plot of the deformable field computed in Figure 4-19 after 600 iterations. (a) Quiver plot of the deformation field using method A, (b) quiver plot of the deformation field using method B. The deformation field by method B is smoother than the deformation field by method A.

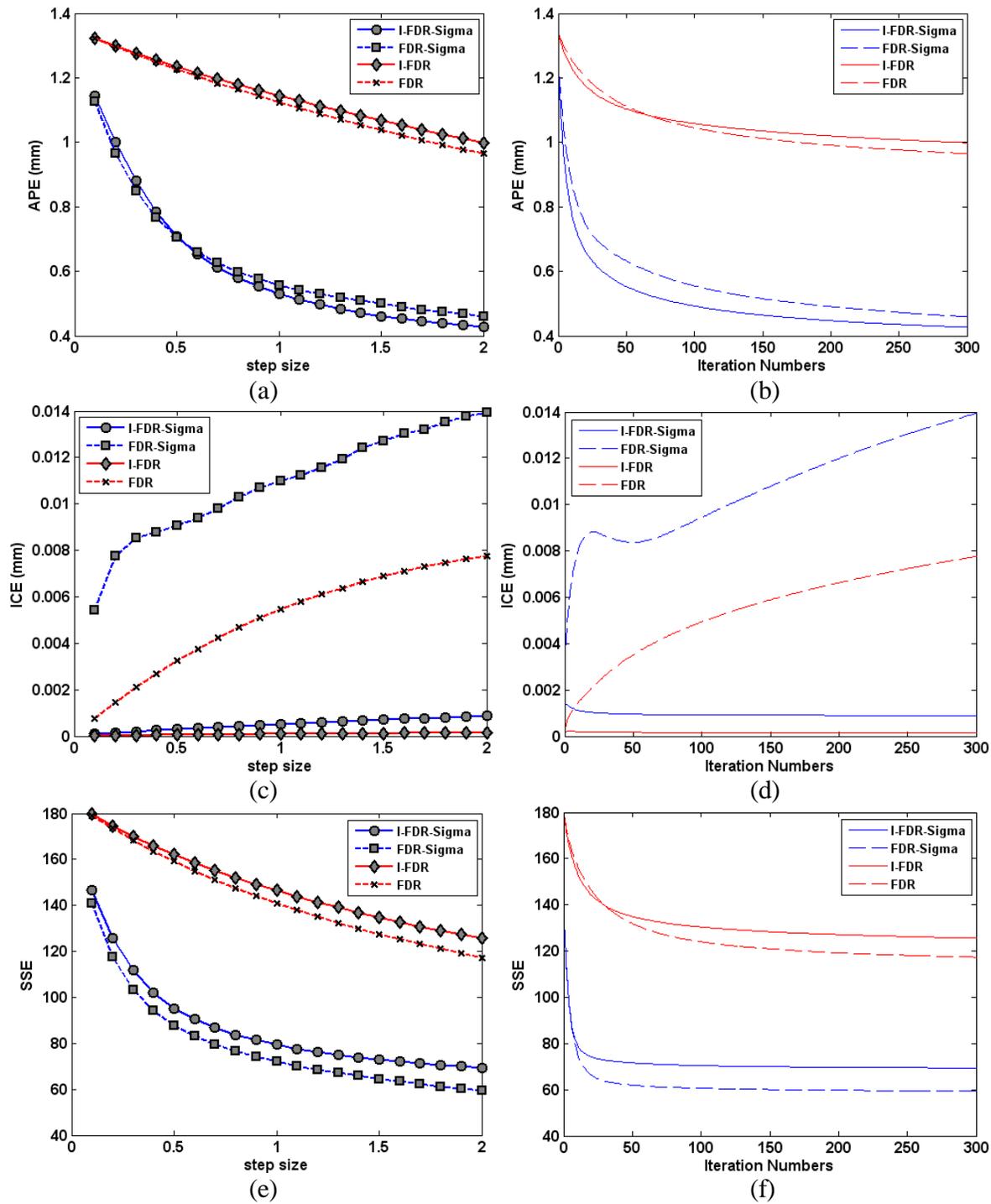


Figure 4-21. The evaluation of the FDR, I-FDR, FDR-Sigma, and I-FDR-Sigma algorithms using high noise myocardium phantom. (a) Change in APE at the 300<sup>th</sup> iteration with step size. (b) Change in APE with the number of iterations at a step size corresponding to the minimum APE. (c) Change in ICE at the 300<sup>th</sup> iteration with step size. (d) Change in ICE with the number of iterations at a step size corresponding to the minimum APE. (e) Change in SSE at the 300<sup>th</sup> iteration with step size. (f) Change in SSE with the number of iterations at a step size corresponding to the minimum APE.

Table 4-1. Change in step size of the FDR-group and Demons-group algorithms for robust measure

		Simulated phantom	Physical phantom	Clinical image
FDR-group	From	0.1	0.1	0.1
	To	2.0	2.0	1.0
	Increment	0.1	0.1	0.1
Demons-group	From	0.1	0.1	0.1
	To	1.0	1.0	0.9
	Increment	0.1	0.1	0.2

Table 4-2. The mean and standard derivation (SD) of APE, ICE, and SSE of seven DIR algorithms using myocardium physical phantom, the image pixel dimension is 1 mm x 1 mm x 1 mm on X, Y, and Z direction.

	APE(mm)		ICE(mm)		SSE	
	Mean	STD	Mean	STD	Mean	STD
Demons	0.58	0.12	1.35	0.69	188.73	45.69
I-Demons	0.52	0.078	0.033	0.019	195.18	47.80
B-Demons	0.60	0.098	0.029	0.025	181.82	42.83
FDR	0.67	0.17	0.0085	0.0020	342.18	160.91
I-FDR	0.64	0.20	0.00027	0.00010	369.20	168.87
FDR-Sigma	0.65	0.10	0.027	0.0093	157.22	44.88
I-FDR-Sigma	0.44	0.041	0.0042	0.0044	220.26	46.76

## CHAPTER 5 CLINICAL APPLICATION OF DEFORMABLE IMAGE REGISTRATION

### 5.1 Auto Re-contouring

Auto re-contouring is an important clinical application of DIR. Auto re-contouring makes it possible to automatically propagate the physician-drawn contours in the planning image dataset to a new image dataset. In 4DCT lung imaging, after auto re-contouring, the planning contours in the planning CT can then be automatically transferred to the phase CT corresponding to the patient breathing cycle. The automatically generated contours can assist monitoring the target change during the course of radiation therapy with minimum human interference.

There are generally two frameworks for re-contouring<sup>28</sup>: 1) the ‘snake’ based framework and 2) the registration based framework. The registration based framework was discussed in the Chapter 1.2, where the DIR is first applied to compute the deformation field between two image volumes, then the planning contours are propagated to the new images by the computed deformation field. The ‘snake’ based framework is indeed the image segmentation techniques<sup>143</sup>, where the iterative algorithm seeks to separate the target by evolving the initial contour using both the image gradient information and contour length. Compared to the registration based framework, the accuracy of the ‘snake’ based framework is not affected by the location of the original contours, however, it has the difficulty to re-contouring the targets whose boundaries are not well visually defined, e.g., CTV or PTV. In contrast, registration based framework is able to propagate all the planning contours to the new image dataset, however, the accuracy of the propagated contours depends on not only the accuracy of the DIR algorithm itself, but also the fidelity of the planning contours. Because of its flexibility, a registration based re-contouring was implemented. In the following, we will discuss the details of the registration based auto re-contouring techniques.

Lu et al.<sup>28</sup> proposed an automatic re-contouring algorithm combining DIR techniques and triangulated surface construction techniques. While our re-contouring approach is similar to the Lu's work, it differs in a number of respects. First, we used I-FDR-Sigma deformable registration algorithm, which is more accurate and converge faster. Second, open source visualization toolkit (VTK)<sup>144</sup> was used to assist surface reconstruction and graphics tasks, such as volume cross section generation. VTK is a software framework for 3D computer graphics, image processing, and visualization. It is under active development and used by thousands of researchers and developers around the world.

The flowchart of the auto re-contouring is shown in Figure 5-1. The auto re-contouring consists of four components: 1) surface reconstruction; 2) deformation field computation; 3) surface propagation; and 4) post processing. Surface reconstruction is to create a 3D triangular planning surface mesh that connects all the existing 2D planar contours outlined by the radiation oncologists in the planning dataset. After generating the planning surface mesh, a deformed surface mesh was created by deforming the planning surface mesh using the computed deformable field. As shown in Figure 5-2, the white cubes represent the nodes ( $A$ ,  $B$ , and  $C$ ) of a triangle from the planning surface mesh. After mesh deformation, the original nodes ( $A$ ,  $B$ , and  $C$ ) are moved to the new locations represented by the yellow cubes ( $A'$ ,  $B'$ , and  $C'$ ) by the deformation field. The mesh deformation does not change the relative connectivity of the neighborhood mesh nodes, but their distance between the connected nodes can vary. The post processing of the deformed mesh, such as the generation of 2D planar contours, can be used for assisting contour visualization for the clinicians. To generate 2D planar contours, a series of parallel planes were used to cut the deformed surface meshes and the auto contours were computed by the intersection between the surface mesh and the parallel planes.

In the following, we will explain necessity to apply surface reconstruction for auto re-contouring. As shown in Figure 5-3, because the deformation exists in the 3D space, after deformation, the planning 3D planar contour points will usually not be on the same plane and form the scattered 3D points. There are two approaches to produce the 2D deformed contours from the scattered 3D contour points: 1) the 3D interpolation based approach and 2) the surface reconstruction based approach. Let us first take a look at the intensity range of the contours. Generally the intensities of the contours are binary, that is, the intensity of the points inside the contour is one and outside the contour is zero. The disadvantage of the interpolation based approach is that it will introduce the pixels with the intensity between 0 and 1, e.g., 0.5. It is, therefore, difficult to determine the location of those pixels, i.e., inside or outside the contours. Figure 5-4 illustrates the problem in the 2D case. In Figure 5-4, the plane 1 and plane 3 have the two contours, where pixels with intensity of one are the points inside or on the contours and pixels with intensity of 0 are the points outside the contours. Here the goal is to compute the contour on the plane 2 located in the middle of the plane 1 and plane 3. The interpolation based approach is illustrated in Figure 5-4 (a), where a linear interpolation scheme is applied to compute the pixels in the plane 2, the linear interpolation scheme could be expressed as

$$P_2 = \frac{(P_1 + P_3)}{2} \quad (5.1)$$

where  $P_2$  is the pixel intensity in the plane 2,  $P_1$  and  $P_3$  are the pixels that have the shortest distance to  $P_2$  in the plane 1 and plane 3 respectively. After interpolation, the pixels in the plane 2 consist of two intensity values, 0.5 or 1. Intuitively the pixels in blue between the dashed lines are considered the points within the contour. In the interpolation based approach, two pixels with intensity of 0.5 were inside the contour, however four pixels with the same intensity of 0.5 were outside the contours. Therefore it is almost impossible to distinguish the pixels by the intensity

value alone for interpolation based approach. In surface reconstruction based algorithm shown in Figure 5-4 (b), the dashed lines represent the reconstructed surface that can be easily used to determine whether a point is inside the contour or outside the contour: the points within the surface are the points inside the contours (in blue) and the points outside the surface are the points outside the contours (in red). Moreover, no pixel has a value between 0 and 1 in this case. Surface reconstruction based approach is always preferred in the auto re-contouring, especially in the cases where the shape of the top and the bottom contours is dramatically different.

### **5.1.1 Surface Reconstruction**

As illustrated in Figure 5-5, the physician-drawn planning contours (Figure 5-5(a)) are exported from the TPS, the surface reconstruction is used to connect the contour points by triangular (the process is also called triangulation), where each edge of one triangle is the edge of two nearby triangles. Surface reconstruction from planar contours is a developed technique in computer vision and computer graphics. Christiansen et al.<sup>145</sup> proposed a limited triangulation algorithm based on the normalized contours points and shortest diagonal principle, however, the mesh quality could be poor for non-convex contours. Here, a robust surface reconstruction algorithm (`vtkVoxelContoursToSurfaceFilter`) available in VTK<sup>144</sup> was used to reconstruction the surface mesh. Figure 5-5 (b) shows the generated surface mesh from the planning contours in Figure 5-5 (a). The close view of the mesh is available in Figure 5-5 (c).

### **5.1.2 Post Processing**

Post processing is to generate layers of 2D deformed contours from the 3D deformed surface mesh. 2D contours will make it easy for the clinicians to compare the contours before and after deformation in the TPS. Two functions in VTK, “`vtkCutter`” and “`vtkPlane`”, were used to create the 2D contours.

The in-house software was developed for auto re-contouring. It was based on the VTK<sup>144</sup>, Matlab (The Mathworks Inc, Natick, MA), and python programming language<sup>146</sup>. The code snippet of surface reconstruction, deformation, and post processing was available in Appendix A. An example of auto re-contouring was shown in Figure 5-6, where the planning contours of GTV (i.e., the green contour) and right lung (i.e., the brown contour) in the planning CT were mapped onto the phase CT. The auto re-contoured right lung (i.e., the red contour) and GTV (i.e., the blue contour) matched the anatomical boundaries. The auto re-contouring is able to compensate for the location and shape change of the target.

## **5.2 Auto Internal Target Volume Generation**

Another clinical application of DIR in 4D treatment planning is auto internal target volume (ITV) generation. In our institute, the ITV is defined as the union of the GTV from different phases of the 4DCT images. Traditionally ITV contouring is a time consuming process because the physicians require manually contouring the GTV in each phase CT. Auto ITV generation only requires the physician to contour at one CT dataset without the cumbersome work of the contouring in each phase CTs. It consists of two steps: first, physician is required to contour the GTV in one phase CT; second, the contoured GTV is automatically propagated to the other phase CT by auto re-contouring and DIR; third, the ITV is computed by the union of the GTV in all phase CTs. Furthermore, a probability density function (pdf) can be computed by assigning the probability to each pixel in the GTV region. In Figure 5-7, an ITV is automatically computed from a 4DCT image dataset consisting of 10 phases CT. The color map represents the probability that a GTV voxel occupies a specific pixel location for all phases CT in the respiratory cycle. The ITV is generated by the union of all the GTV in the phase CTs.

The auto ITV generation provides an effective treatment planning tools for 4DCT without increasing the overall physician workload associated with manual segmentation of structures on each phased CT within the 4DCT.

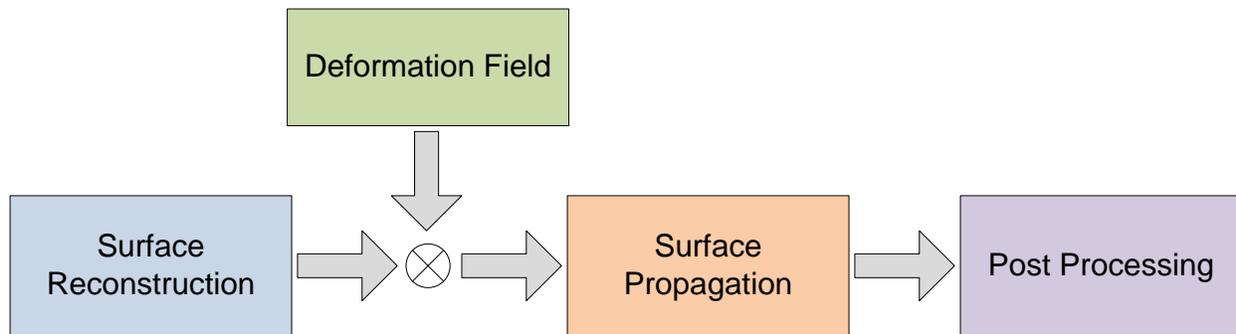


Figure 5-1. Flowchart of the auto re-contouring. The 3D surface mesh is reconstructed from the 2D layers of the manually contoured structures. After applied deformation field computed from DIR, the 3D surface mesh was deformed and represented the anatomical change. The post processing, such as 2D planar contours generation, will assist the visualization of the deformed structures.

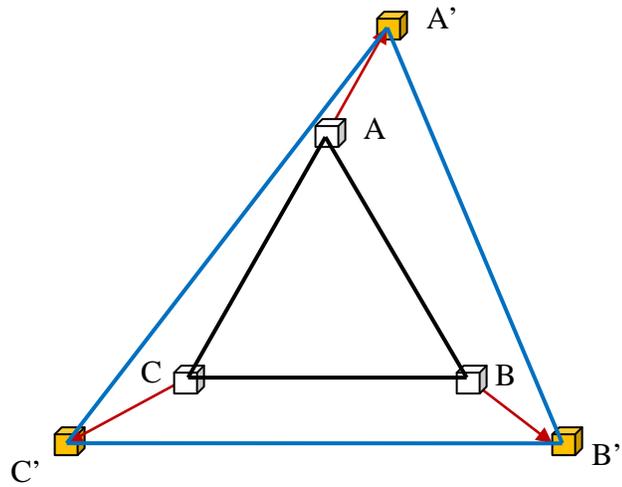


Figure 5-2. Mesh deformation. The white cubes represent the nodes (A, B, and C) of a triangle from the planning surface mesh. After mesh deformation, the original nodes (A, B, and C) are moved to the new locations represented by the yellow cubes (A', B', and C') by the deformation field. The mesh deformation only changes the distance between the nodes, without affecting their connectivity.

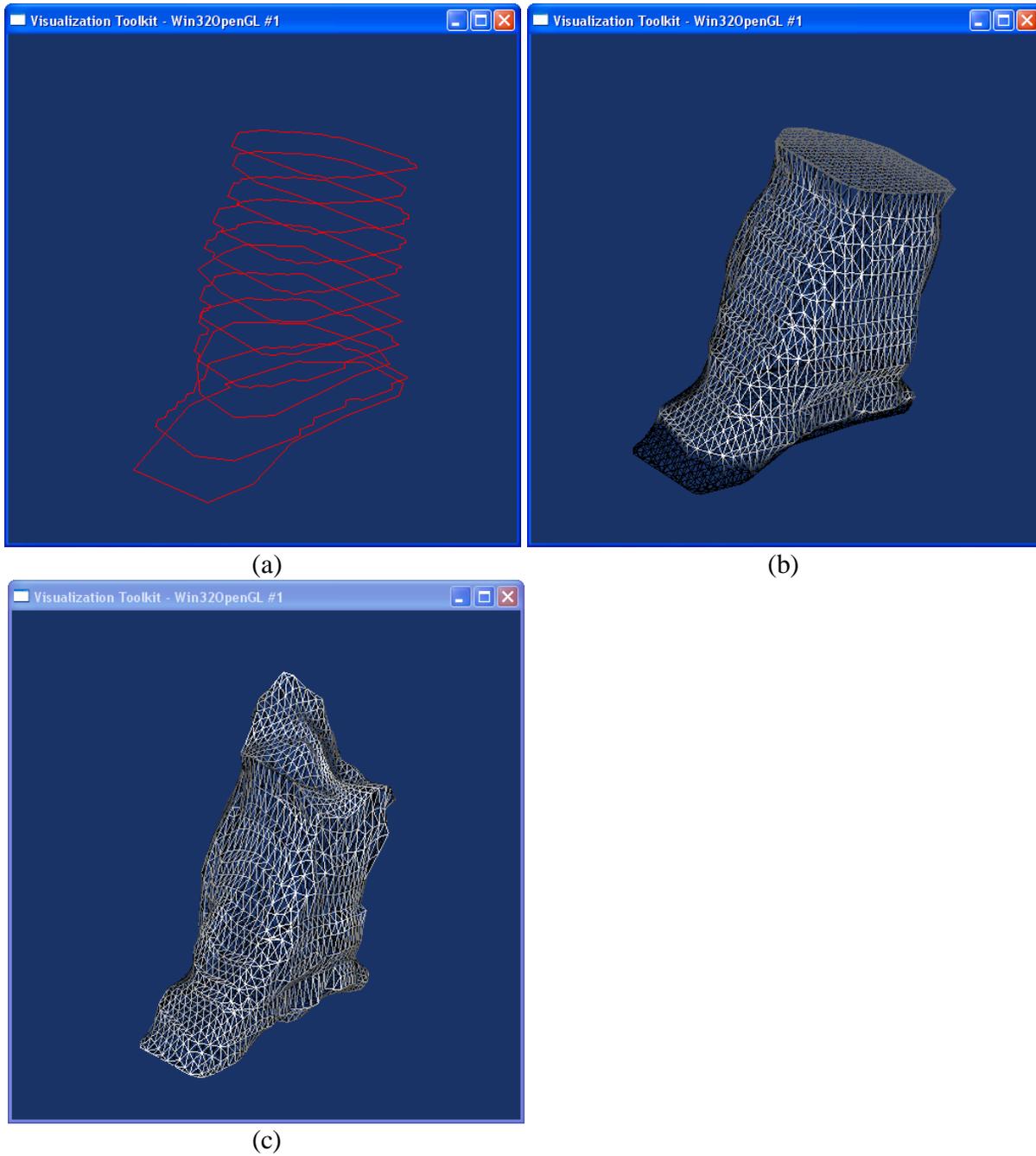


Figure 5-3. Surface reconstruction and surface mesh deformation. (a) The planning GTV contours. (b) Surface mesh after using surface reconstruction algorithm. (c) Surface mesh deformation after applying deformation field.

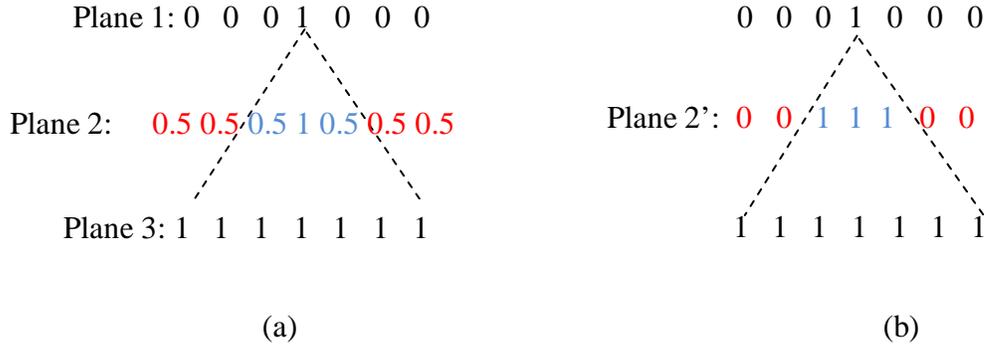


Figure 5-4. A 1D example of interpolation based re-contouring and surface reconstruction based re-contouring. The contour in the middle plane (plane 2) is needed to be generated based on the known contours in the plane 1 and the plane 3. (a) Interpolation based re-contouring has difficulties to determine whether the pixels in plane 2 is inside the contours because of the same pixel intensity for points inside or outside the contours; (b) Surface reconstruction based re-contouring can generate the contour in plane 2' by determining whether the pixel is inside the reconstructed surface.

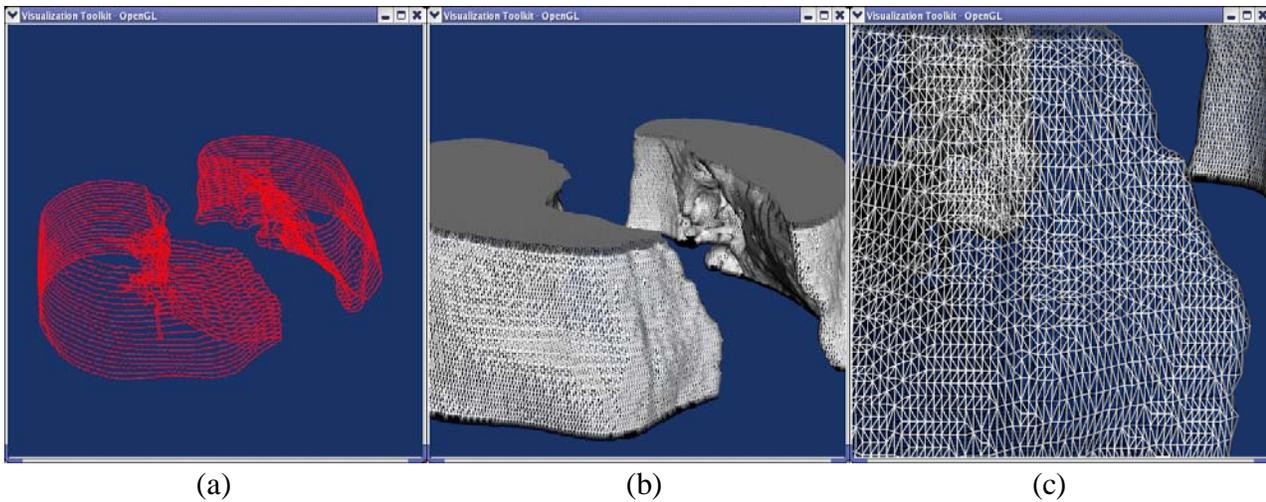


Figure 5-5. Surface reconstruction from 2D planar contours. (a) Visualization of layers of planning contours drawn by the physicians; (b) Reconstructed surface connecting the contour points using triangular mesh; (c) Close view of the triangular mesh.

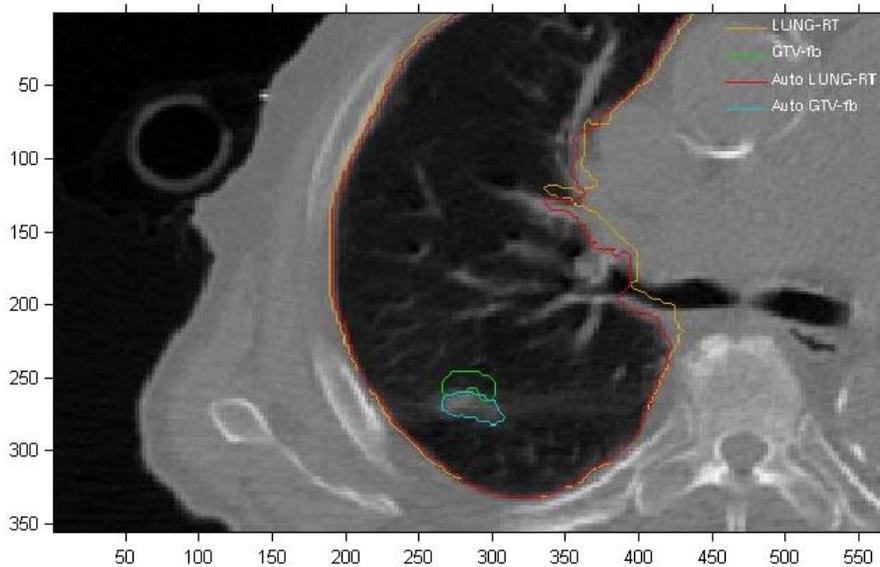
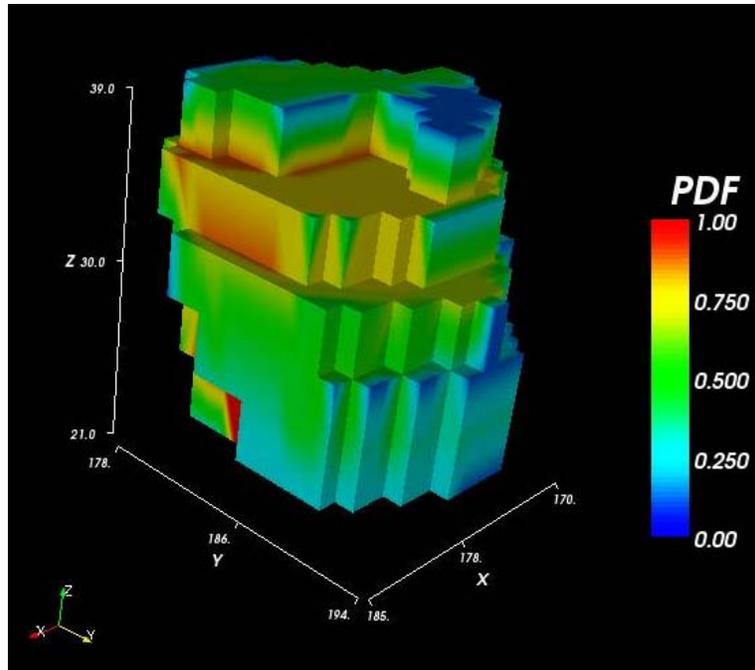
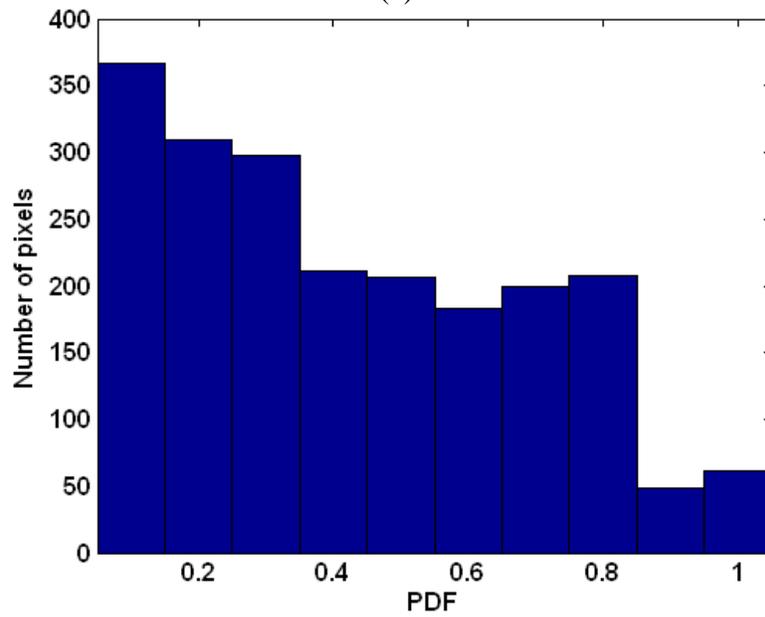


Figure 5-6. Auto re-contouring for lung image using DIR. Physician drawn contours (drawn on separate source CT) and computer generated contours, using auto re-contouring, have been overlaid on a target CT. This zoom-in view has the dimension of 358 x 355 with the pixel dimension of 0.4 mm by 0.4 mm. The LUNG-RT and GTV-fb (brown and green contours), representing outlining the right lung and gross tumor respectively, are physician drawn contours on the source CT. The Auto LUNG-RT and auto GTV-fb (red and blue contours) are the corresponding computer generated contours on target CT based on DIR.



(a)



(b)

Figure 5-7. ITV generation. (a) The color map represents probability density function, which indicates the probability that a GTV voxel occupies a specific pixel location for all phases in the respiratory cycle. The ITV is the union of all nonzero probability voxels. The pixel dimension is 1 mm x 1 mm x 3 mm. (b) The histogram of the probability density function.

## CHAPTER 6 NEAR REAL-TIME DEFORMABLE IMAGE REGISTRATION USING GRAPHICS PROCESSING UNITS

As discussed in the previous chapters, DIR plays an important role in estimating target motion and monitoring the target evolution. However, existing DIR algorithms are computational intensive, typical reported computation time of CT images was: 5 minutes for image volumes of 128 x 128 x 128 pixels using demons<sup>30</sup>, 3 minutes for image volumes of 256 x 256 x 61 pixels using free-form deformation<sup>31</sup> on Pentium III 933MHz personal computer (PC), 6 minutes for image volumes of 256 x 256 x 61 pixels using accelerated demons<sup>22</sup> on Pentium 2.8GHz PC. Therefore, for a typical image size of 256 x 256 x 100 pixels, one can expect the registration time be at least 5 minutes using the standard algorithms. One should expect more computation time if considering additional analysis for clinical applications in ART, such as auto re-contouring<sup>147, 148</sup>, dose recalculation<sup>149, 150</sup>, and treatment re-planning<sup>150, 151</sup>. For the clinical usage of DIR, a desirable computation time is within two to three minutes. To meet the clinical demands, researches have been carried out using high performance computing (HPC) techniques to accelerate the computation of DIR algorithms. In the following, we will first review the current high performance computing techniques in image registration using different hardware architectures, and then a proof of concept study will be presented to demonstrate the feasibility of the near real-time Demons algorithm. The Demons algorithm, as already discussed in Chapter 4, though not necessarily the fastest or most robust in all clinical cases, is widely available and serves as a standard for comparison due to its simplicity and efficiency. The HPC implementation of some essential image processing components, such as image gradient and convolution, can assist future research in accelerating DIR using HPC.

## **6.1 Review of High Performance Computing In Medical Image Registration**

To improve the performance of image registration, HPC techniques have been studied using different hardware architectures. The following HPC architectures will be reviewed including 1) field-programmable gate array (FPGA); 2) cell broadband engine (CBE) processor; 3) supercomputer or clusters; 4) graphics processing unit (GPU).

### **6.1.1 Review of Image Registration Using FPGA**

FPGA is a programmable silicon chip utilizing the prebuilt logic blocks and programmable routing resources. FPGAs can be configured to implement customized hardware functionalities without soldering irons. To implement algorithms in FPGA, algorithms are firstly developed in the design software, then compiled as the configuration file and downloaded to the FPGA. The availability of the high-level design tools, such as graphical block diagrams or interface to C language, makes the FPGA programming convenient.

FPGA techniques were used in image registration to accelerate computation of mutual information. Based on an Altera Stratix II EPSS180F1508C4 FPGA (Altera Corporation, San Jose, CA), Dandekar et al.<sup>152-154</sup> demonstrated a FPGA architecture for multi-modality registrations of abdominal intra-procedural non-contrast CT with pre-procedural contrast-enhanced CT and PET images. The registration was based on the hierarchical volume subdivision-based DIR, where the image volume was subdivided into small sub-volumes, rigid registration was applied to each sub-volume, and the final deformation was computed by combining locally rigid registration of the sub-volumes with quaternion-based interpolation. In their study, the FPGA played a role of accelerating the mutual information computation in the registration. Comparing with the C++ implementation on an Intel Xeon 3.6 GHz PC, the FPGA provided a speedup of 41 for an image size of 256x256x256 pixels and a speedup of 21 for an image size of 16x16x16 pixels. Castro-Pareja et al.<sup>155, 156</sup> investigated the FPGA application of

accelerating mutual information calculation, where the fast automatic image registration (FAIR) architecture was proposed to improve the mutual information calculation by accelerating the histogram computation. The FAIR system was based on an Altera Stratix EP1S40 FPGA and achieved a speedup of 86 comparing with the sequential code on a Pentium III 1GHz PC.

Jiang et al.<sup>157</sup> proposed an FPGA-based design supporting free-form deformation using third order B-spline model on 2D images. The approach included pre-calculating the B-spline basis function, transforming a nested loop to avoid conditional calculation, and developing fully pipelined circuits and multiple pipeline designs. the FPGA implementation, based on a Xilinx XC2V6000 FPGA at 67 MHz,, run 3.2 times faster than an Intel Xeon 2.6 GHz PC.

It was pointed out that the current FGPA implementation cannot execute the whole registration task independently; it had to communicate with the host machine and served as an additional computation unit. To the best of our knowledge, no complete FPGA system, which included all the necessary registration components, was reported.

### **6.1.2 Review of Image Registration Using Cell Broadband Engine (CBE)**

CBE was developed by the collaboration of Sony, IBM, and Toshiba. The Cell objective was to achieve 100 times performance of the PlayStation2 and led way for the future<sup>158</sup>. The first generation of CBE combined a 64-bit power processor element (PPE) with eight synergistic processor elements (SPEs). Each SPE had a SIMD (Single-instruction, multiple-data) unit, which could perform a floating or an integer operation on four data elements at each clock cycle. The PPE, on the other hand, was more general processor and was capable of running a conventional operating system. It could also control the SPEs and could start, stop, interrupt, and schedule processes running on the SPEs. To achieve high performance for computational intensive tasks and reduce the memory latency, EIB (Element Interconnect Bus) was introduced to provide high

bandwidth access to the main memory and the external data storage. Readers can refer to <sup>158</sup> for the details of CBE.

After the debut of the first generation of CBE in 2005, studies were carried out utilizing CBE in medical image registration. Ohara et al. <sup>159</sup> presented a scheme of implementing mutual information based multi-resolution 3D image registration on an IBM BladeCenter QS20, which contained two CBE processors at 3.2 GHz with 1GB memory. To accelerate the image registration, the optimization of the parallelism was exploited in the following: 1) Task parallelization and partition. The program was partitioned into multiple tasks to be fit into the local memory of each SPE. It could be viewed as a distributed memory multiprocessor with a small local memory attached to a large shared memory. 2) SIMD pipeline optimization. SPE was highly pipelined and optimized for SIMD instructions. It was essential to not only perform multiple operations by SIMD instructions but also optimize the code to exploit the architecture. 3) Optimizing memory bandwidth. Two data partition techniques, localized sampling and speculative packing, were applied to optimize the memory traffic and avoid low efficient pixel-wise memory access. Comparing with the sequential version on Intel Xeon 3.0 GHz processor with 4G memory, the CBE run about 11 times faster and was able to reduce the memory traffic by 82%. As a result, it could register a pair of 256x256x30 pixels images in one second using multi-resolution method. However it was unclear in this paper whether the 3D image registration was the rigid registration or the deformable registration. Some registration details, such as levels of the resolutions, the number of bins in mutual information calculation, and number of iterations in each resolution, were not specified in the paper.

### **6.1.3 Review of Image Registration Using Supercomputers or Clusters**

Parallel computers and supercomputers <sup>160</sup> are the traditional hardware platform for parallel computing. The common parallel computers refer to either SMP (symmetric

multiprocessor) servers, which have many symmetrical processors on a single mother board, or computer clusters, which consist of many of the similar type of machines and tightly-coupled using dedicated network connection. The supercomputer, on the other hand, has several different architectures: 1) MPP (Massively Parallel Processor). The MPP consists of large number of processors connected by a high performance network. 2) PVP (Parallel Vector Processor). PVP consists of many specific vector processors and is more expensive than MPP.

Memory architectures described how each processor access the physical memories. It can be categories into two groups: shared memory (Figure 6-1 (a)) and distributed memory (Figure 6-1 (b)). In the shared memory architecture, each processor accesses the same global memory resource; any memory modifications by one processor are visible to all the other processors. The main advantages of the share memory architecture are 1) the global memory access provides easy programming perspective; 2) fast data sharing due to the fast communication between the global memory and its adjunct processors. However, the share memory architecture suffers from the lack of scalability between memory and processors and the potential high cost for designing shared memory machine with the increasing numbers of processors. In the distributed memory architecture, the memory is associated with individual processors and each processor is only able to address its own memory. The main advantages of the distributed memory architecture are 1) no bus or switch to interfering the data access, each processor can utilize the full bandwidth to its own local memory; 2) no inherent limit to the number of processors and sizes of the systems. The main drawbacks in distributed memory system are 1) the difficulty of the inter-processor communication; 2) the difficulty of mapping the existing global memory based data structures into the distributed memory data structures.

Several studies had been carried out using parallel computing techniques to speed up the image registration. Christensen et al.<sup>142</sup> implemented the elastic and fluid deformation image registration algorithms on a MarPar 128x128 massively parallel SIMD computer. For registration 128x128x100 pixels volumes with 100 iterations, parallelized fluid registration took 1.8 hours, about 178 times speed up comparing with a SGI MIPS 150 MHz R4400 computer. Rigid and deformable registration on the shared memory multiprocessor system had also been studied<sup>161</sup>.<sup>162</sup>. Rohlfing and Maurer<sup>162</sup> presented a parallel implementation of a B-spline based free-form deformable registration algorithm on a 128-CPU shared memory system (SGI Origin 3800). Using multithreading programming by partitioning data and tasks, the parallel implementation achieved a speedup factor of more than 50 times and was able to complete the intra-operative brain deformation in less than a minute using 64 CPUs. Wachowiak<sup>161</sup> proposed a parallel optimization algorithm based on the global optimization method called DIRECT (Dividing Rectangles) and the local optimization method called MDS (Multidirectional Search). The parallel registration implementation was performed on a SGI Altix 3000 shared memory system with 20 Intel Itanium-2 CPUs with 1.3 GHz and programmed with OpenMP<sup>163</sup>, which is an application programming interface (API) for shared memory multiprocessing programming. Computer clusters were reported to improve the performance of the medical image registration<sup>164-166</sup>. Warfield<sup>164</sup> implemented a parallel feature-based rigid image registration on a cluster consisting of two Enterprise Server 5000S, each has 8 167 MHz UltraSPARC-I CPUs. Using POSIX and MPI (Message Passing Interface) parallel programming tools, the registration execution time was within 5 -10 minutes. For rigid multi-modality image registration, ourselin<sup>165</sup> proposed an scheme to parallelize block-matching registration algorithm using a cluster with 10 biopro Pentium III 933 MHz PCs. MPI was utilized to coordinate the different

nodes of the cluster, the parallelization of each node was programmed by OpenMP. For small and medium resolution images, the parallelized registration achieved 19 seconds and 45 seconds on five bi-processors respectively. It took 70 seconds for ten bi-processors. Ion<sup>166</sup> presented a performance study of implementing a B-spline based deformable registration on three different clusters. The implementation involved the data-parallel processing technique and the precomputation technique, it was demonstrated that the registration time was reduced from ten hours to ten minutes.

#### **6.1.4 Review of Image Registration Using Graphics Processing Unit (GPU)**

A Graphics Processing Unit (GPU) is a highly parallel computation unit with tremendous memory bandwidth and computational horsepower, but it lacks the programming flexibility of the CPU. GPU computing is now emerging as a viable and high performance alternative for general non-graphics computing tasks, such as DIR. The GPU's substantial arithmetic and memory bandwidth capabilities, coupled with its recent addition of user programmability, has allowed for "general-purpose computation on graphics hardware" (GPGPU). Many non-graphics oriented computationally expensive algorithms have been implemented on the GPU. Developers prefer GPUs over other alternative parallel processors due to several advantages including their low cost and wide availability. Owens et al.<sup>167</sup> presented a comprehensive survey of latest research in GPU computing.

Traditional GPU programming languages have been primarily used for graphics applications, they include OpenGL<sup>168</sup>, DirectX<sup>169</sup>, Cg<sup>170</sup>, and HLSL<sup>171</sup>. New GPU programming languages, such as CUDA<sup>172</sup>, CTM<sup>173</sup>, and Brook<sup>174</sup>, have emerged to allow for efficient programming for general purpose computing. GFLOPS (i.e.  $10^9$  floating point operations per second) is an accepted overall approximate representation of the computational capability of a processor. Current high end processors include the Intel Dual Core™ Xeon 3.0 GHz (4 MB L2

cache, 1333 MHz FSB) (CPU) rated up to 38.3 GFLOPS<sup>175</sup>, and the NVIDIA GeForce 8800 GTX (GPU) rated up to 520 GFLOPS<sup>176</sup>. The challenge for fast DIR using GPU is to implement algorithms in a form that can take advantage of the highly parallel nature of GPU computing.

Since medical imaging applications intrinsically have data-level parallelism with high computational requirements, they are very suitable for GPU implementation. Researchers have conducted several prior GPGPU studies in the field of image registration. However, only a limited number of previous studies concentrate on the DIR.

The current literatures have not paid sufficient attention to GPGPU use in DIR. Instead, GPUs have been mainly used for speeding up the generation of the digitally reconstructed radiograph (DRR) in 2D-3D image registrations<sup>177-179</sup>. On the one hand, image registration is computationally expensive. On the other hand, image registration is high parallel. Hence, GPUs with their high-performance parallel processing power provide great opportunities for speeding up image registration. Large 3D data sets used in DIR algorithms put an even heavier burden on computational resources. However, as indicated by Vetter et al.<sup>180</sup>, there has not been much research into the use of GPU in DIR. Likewise, Sharp et al.<sup>181</sup> do not consider prior work on the GPU-based deformable registration as “well established”. One of the early studies of deformable registration utilizing GPU could be traced back to Soza et al.<sup>182</sup>. They used 3D Bezier functions for non-rigid alignment of respective volumes and utilized the tri-linear interpolation capabilities of GPU to accelerate the deformation of the moving volume. In order to better understand the brain shift phenomenon, Hastreiter et al.<sup>183</sup> performed DIR based on piecewise linear transformations proposed in an earlier study<sup>184</sup> to approximate nonlinear deformations, and accelerated tri-linear interpolation by using the 3D texture mapping capabilities of graphics hardware. Levin et al.<sup>185</sup> exploited graphics hardware to implement a high-performance

thin-plate spline (TPS) volume warping algorithm that could be used in iterative image registrations and accelerated the application of the TPS nonlinear transformation by combining hardware-accelerated 3D textures, vertex shaders, and tri-linear interpolation.

Strzodka et al.<sup>186</sup> implemented a non-rigid regularized gradient flow registration on the GPU to match 2D images. In contrast to earlier studies that used graphics hardware only for the tri-linear interpolation (e.g. Soza et al.'s study<sup>182</sup>), Kohn et al.<sup>187</sup> implemented the entire 3D regularized gradient flow on a GPU. However, due to memory bottlenecks, they reported that their 3D non-rigid registration is not as fast as one would expect. Vetter et al.<sup>180</sup> implemented non-rigid registration on a GPU using mutual information and the Kullback-Leibler divergence between observed and learned joint intensity distributions. They proposed this method for specifically matching multi-modal data sets and, like Kohn et al.<sup>187</sup>, they implemented the entire registration process on the GPU.

To the best to our knowledge, there are only three studies that mapped the Demons registration algorithm to the GPU: implementation of Kim et al.<sup>188</sup>, Courty and Hellier<sup>189</sup>, and Sharp et al.<sup>181</sup>. All these three studies were conducted concurrently and independently with our study and differ mainly on the method they used for smoothing the displacement field. These implementations also used different GPUs and programming environments.

Kim et al.<sup>188</sup> implemented the Demons algorithm using a simple ramp smoothing filter and a NVIDIA GeForce 6800 GPU with the Cg language. They took the average of the closest six neighbors of each voxel to smooth the displacement field. Courty and Hellier<sup>189</sup> presented a GPU implementation of the Demons algorithm using a Gaussian recursive filter. The advantage of recursive filtering is that number of operations is bounded and independent of the standard deviation of the Gaussian filter, but the error will be significant for the standard deviation larger

than 10. In order to implement the recursive filter, the Gaussian filter with 4<sup>th</sup>-order cosines-exponential functions was approximated. In this implementation, they chose an NVIDIA Quadro FX 1400 GPU, with fragment programs written with a Cg-like syntax. Finally, Sharp et al.<sup>181</sup> implemented the Demons algorithm using a separable Gaussian filter. They used the Brook programming environment and a NVIDIA GeForce 8800 GTS GPU. Using a Gaussian filter for smoothing the displacement field provides the most accurate implementation. However, this is the most expensive part of the Demons algorithm. Hence, to achieve faster runtimes some of the researchers simplified or approximated the smoothing process. For instance, Kim et al. used simple ramp smoothing instead of Gaussian smoothing and Courty and Hellier approximated the Gaussian filter as discussed above. Sharp et al. was first to use a separable Gaussian filter. In order to provide an accurate implementation, we have also used Gaussian smoothing and performed convolution of the displacement field with separable Gaussian filter like Sharp et al. did. However, we improved the speed of Sharp et al.'s implementation by using NVIDIA's CUDA environment instead of Brook; we report a 10% faster runtime on the same hardware. CUDA specifically targets newer NVIDIA cards and provides powerful features, such as shared memory access, which we extensively used in our implementation that Brook does not offer. Hence, CUDA is optimized for these cards and provides better support/performance than Brook. In addition, Brook is a legacy academic project and is at best maintenance-mode-only, whereas CUDA is directly supported by NVIDIA, is under active development, and has a broader set of programming tools and libraries available. In addition to achieving faster runtime, we obtained constant speedups across datasets with different image sizes. Kim et al. and Courty and Hellier reported speedup for only one dataset. Sharp et al. reported speedups for two datasets. However, they reported a smaller speedup for the larger dataset size. Hence, to the best to our knowledge,

our implementation presents the fastest and most scalable GPU-based implementation of Demons algorithm available to date.

It has to be noted that current GPUs are limited by single precision floating point (i.e. 32 bits), while CPUs can use double precision floating point (i.e. 64 bits). Typically single precision is deemed to be sufficient for DIR in medical imaging, and other general computations, including diffusion equations, wave equations, Poisson problem, and Navier-Stokes equations<sup>190</sup>. However there are cases, such as for finite element modeling, where double precision is needed<sup>154</sup>. As well, significantly less dedicated processor memory is available to GPU programming than for CPU programming. Currently GPUs typically have less than 1 GB memory. For CPUs up to 4GB RAM is available with the 32-bit architecture, and theoretically up to  $1.7 \times 10^{10}$  GB RAM with the 64-bit architecture.

In the next section, we present the implementation of the demons algorithm for the near real-time DIR based on a commercial GPU for a standard PC architecture and CUDA programming language. We consider the central issue of GPU vs. CPU computing: the effect of the computational organization, including data structures, and structure of parallelism on computing performance. A systematic comparison of GPU and CPU computing strategies for DIR in terms of computational efficiency time, and cross-correlation (CC) between the deformed images and the target images is presented. For this relatively nascent and highly promising technology, we consider a standard and widely disseminated DIR algorithm: Demons. The details of the Demons algorithm are discussed in the Chapter 4.2. The Demons algorithm, though not necessarily the fastest or most robust in all clinical cases, is widely available and serves as a standard for comparison due to its simplicity. This initial choice will permit other researchers to standardize performance comparisons and build on our promising results. We evaluate this

strategy over a range of clinical lung imaging volumes acquired with 4DCT. In contrast to the previous studies, which only presented single-threading CPU implementation runtimes, we will provide multi-threading CPU implementation runtimes, in order to present a fair comparison between CPU and GPU speeds.

## 6.2 Materials and Methods

Lung and prostate imaging are candidates for clinical applications of DIR, requiring fast near real-time DIR if one is to use time series volumetric imaging data as part of ART. We evaluated GPU DIR using clinical CT lung imaging acquired with 4DCT from eight patients. CT images were acquired using a Phillips BRILLIANCE 16 slice CT-Simulator (Phillips Medical Systems, Cleveland, Ohio). CT data was binned over the 10 phases of a respiratory cycle, the free breathing (FB) CT represented an average over the respiratory cycle sampled from the 10 phases. The FB CT served as the source CT, since at our institution, by convention and historical consistency, all segmentation is initially carried out on the FB CT. The target consisted of one of the ten phases, typically with the largest deformation. The Demons algorithm was used to map the FB CT data set onto a phased CT. It was implemented on an Intel dual core 2.4GHz CPU using C language, and on a NVIDIA 8800 GTX GPU using CUDA. In our implementations, the available dedicated RAM for the CPU was 3GB, while the video memory for the GPU was 1536MB (The NVIDIA 8800 GTX with 1536MB used here is equivalent to the NVIDIA Quadro FX 5600). The following parameters were used for the demons algorithm: size of smoothing kernel is 3 x 3 x 3 pixels, standard deviation of smoothing kernel is 0.5, and step size = 1.0.

The dual core CPU implementations consisted of single threading (i.e. equivalent to single CPU processing) and multithreading (i.e. equivalent to dual CPU processing). Current GPUs are capable of single precision floating point operation (i.e. 32 bit processing), while CPUs allow for double precision floating point operation (i.e. 64 bit processing). The cross-correlation (CC)

metric was used to quantify the resulting difference image for DIR from GPU and CPU implementations. In the study, we compared the performance of the CPU and the GPU implementation using the following criteria: TPI (time per iteration) and TPMI (time per megapixels per iteration). TPI represents the computation time per iteration while TPMI represents the computation time per million pixels per iteration. TPI is independent of the termination criterion, and typically varies linearly with image size. TPMI is a normalized measure (with respect to data size) of TPI, and is a better measure of computational efficiency. The variation of TPMI expressed as a percentage of the mean (i.e.  $100\% \times \text{standard deviation (sd)} / \text{mean}$ ) provides a measurement of sensitivity of computational efficiency to data size. Ideally TPMI should be the same regardless of the image size. This is typically the case for CPU computing, with the possible exception due to very large data sets that cannot be entirely stored on RAM but are frequently accessed by CPU. Invariance of TPMI with respect to data size is desirable and can indicate programming efficiency. The maximum TPMI, mean TPMI, and standard deviation of the mean TPMI provide a test for this invariance over the imaging volumes encountered clinically. Since there was no absolute gold standard for the clinical imaging, the CC between the deformed and the target image, and the difference image were used to evaluate the registration results between CPU DIR and GPU DIR.

### **6.3 Results**

It was observed that 100 iterations were sufficient to achieve good convergence for DIR in all 8 clinical cases. The termination condition was empirically set that the fractional difference in similarity measures (i.e. cross correlation) between consecutive iterations was less than  $10^{-3}$ . The source image, target image, deformed image, and the difference images are shown in Figure 6-2. As well, no difference between the GPU and CPU based deformation was visually observed in the difference images. The effect of single precision float computing (GPU) and double precision

floating point (CPU) on the computation of CC is quantified in Table 6-1. Even with different programming architectures, the difference was minimal and was not significant to the deformation mapping. For CC, a mean difference of  $4 \times 10^{-6}$  and a maximum difference of  $2 \times 10^{-5}$  were observed between GPU and CPU computations over 100 iterations for the 8 clinical cases. This was not statistically significant, as it was smaller than the termination criteria. In fact, different programming implementations of Demon's algorithm (i.e. MATLAB vs. C) also produce small but clinically insignificant differences measured by the cross-correlation and the difference image.

For the clinical CT images, using DIR based on 100 iterations and the demons algorithm, GPU time was in the range of 1.8 - 13.5 seconds for data sizes of  $2 \times 10^6$  -  $4 \times 10^7$  pixels (representative of imaging volumes in radiotherapy), which we considered to be near real-time. In Table 6-2, the individual CPU and GPU time were reported for each clinical data set. TPI for GPU ranged from 0.0181-1.35 seconds, compared to 1.03-7.45 seconds for single thread CPU and 0.671-4.66 seconds for dual thread CPU. Normalizing for image size, the TPMI for GPU was 0.00916 seconds, while it was 0.527 seconds and 0.335 seconds for single thread and dual thread CPU, respectively. Hence the GPU was 55-61 times faster (based on ratio of TPIs or TPMIs) than the single threading CPU, and 34-39 times faster than multithreading CPU. The multithreading implementation was efficiently optimized to take advantage of the dual core architecture since the ratio of multithreading to single threading applications was approximately 0.63 for both TPI and TPMI measurements. A value of 0.5 is considered ideal. It should be further noted that the current impressive GPU implementation involved computing the CC value at each iteration, without CC evaluation, the GPU time could be further reduced by approximately 10%. CC is not required for the demons algorithm but we implemented in order to

study any CPU and GPU computational differences. The improvement in computational efficiency using the GPU, and the linear dependence of TPI (for both CPU and GPU) on image size were shown in Figure 6-3.

As expected, for both single threading and multithreading CPU implementations, the TPI is a function of image size but TPMI is almost invariant of image size. Similar behavior was also observed for GPU. The standard deviation in TPMI expressed as a percent of mean (i.e.  $100\% \times \text{sd} / \text{mean}$ ), was 1.3% for single thread CPU, 1.6% for dual thread CPU and 3.7% for GPU using CUDA. This indicated very good memory management on the part of CUDA.

In Table 6-3, for the purpose of comparison, we have listed the GPU performance measured based on TPI and TPMI of the implementation by Sharp et al.<sup>181</sup>, who carried a comparison between GPU based DIR using a NVIDIA 8800 GTS GPU and CPU DIR (using a 2.8 GHz Intel dual core CPU and 1.5GHz RAM). Our implementation had TPMI of  $8.75 \times 10^{-3}$  seconds for the image size of  $256 \times 256 \times 64$  ( $=4.19 \times 10^6$  pixels), while Sharp had a value of  $10.4 \times 10^{-3}$  seconds for a similar image size ( $256 \times 128 \times 128$ ), resulting in a 16% improvement. For the larger image size of  $256 \times 256 \times 175$  ( $=11.47 \times 10^6$  pixels), we obtained a TPMI of  $8.72 \times 10^{-3}$  seconds, while Sharp's implementation had TPMI of  $12.16 \times 10^{-3}$  seconds using an image size of  $424 \times 180 \times 150$  ( $=11.45 \times 10^6$  pixels), resulting in an improvement of 28%. These results were presented in Figure 6-4. The differences in the performance of the two implementations could be due to different hardware (NVIDIA 8800 GTX vs. NVIDIA 8800 GTS), programming language (CUDA vs. Brook), and possibly slightly different implementations of the demons algorithm. We did not carry out an explicit comparison of performance based on the CUDA and Brook programming languages using the same GPU hardware. However, an examination of the standard deviation of TPMI seems to suggest that our implementation of CUDA may be better at

memory utilization than Brook. For our GPU implementation, our TPMI had a standard derivation of 3.7% of the mean, while that of Sharp et al. had a standard derivation of 10.9% of the mean. A lower standard derivation suggests less dependence of computation efficiency (normalized with respect to image size) on image size.

#### **6.4 Discussion and Conclusion**

GPU computing offers fast parallel computing within the PC architecture, and has been shown to be highly promising for the DIR in clinical applications. However, compared to CPU programming, GPU programming is challenged by the difficult programming environments, single floating point precision, and reduced dedicated processor memory. A GPU implementation of DIR utilizing the demons logarithm and CUDA language was presented for clinical CT lung imaging. Compared to the traditional CPU implementation, there was negligible difference in the registration quality based on CC value during 100 iterations. GPU registration time based on 100 iterations, which was more than sufficient to achieve desired convergence, were 1.8-13.5 seconds for the clinical data sets with image sizes from  $2 \times 10^6$  to  $1.4 \times 10^7$  pixels encountered in radiotherapy. The GPU was 55-61 times faster than the CPU with the same registration algorithm implemented with C language for single threading implementation, and 34-39 times faster than CPU using multithreading implementation on dual core 2.4GHz CPU. Thus, by using a relatively simple algorithm, such as the Demons algorithm, GPU based image registration yielded near real-time performance. The computation efficiency, normalized for image size, was characterized by TPMI (time per megapixels per iteration) and its standard deviation. The TPMI for GPU was 0.00916 seconds, significantly less than that for CPU (single threading - 0.527 seconds, multithreading - 0.335 seconds). The standard deviation of TPMI, listed here as a percent of mean (i.e.  $100\% \times \text{sd}/\text{mean}$ ), was only 3.7%, indicating that the CUDA programming achieved highly parallelized computing for the indicated data sizes. The

GPU computational efficiency presented here shows a conservative estimate, with a further reduction of approximately 10% in GPU computation time for both TPI and TPMI achieved by not requiring CC evaluation, which was not necessary for the demons algorithm but to study the CPU and GPU computational differences.

Although only Demons algorithm was considered here, GPU computing potentially offers significant performance acceleration for other algorithms. For example, similar improvement can be expected for the “accelerated Demons” algorithm<sup>22</sup>, which applies the ‘active force’ concept to both the source and target images (conventional demons applies to only one image volume), and represents no different function type evaluations for GPU or CPU computing. Future work will consider DIR algorithms with even faster convergence and improved robustness than the Demons algorithm considered here. Furthermore, DIR algorithms are expected take advantage of improved GPU hardware, which historically has seen faster growth in computational power compared to CPU hardware. The possibility of GPU based real-time DIR opens up a host of clinical applications, for ART and other areas in medical imaging.

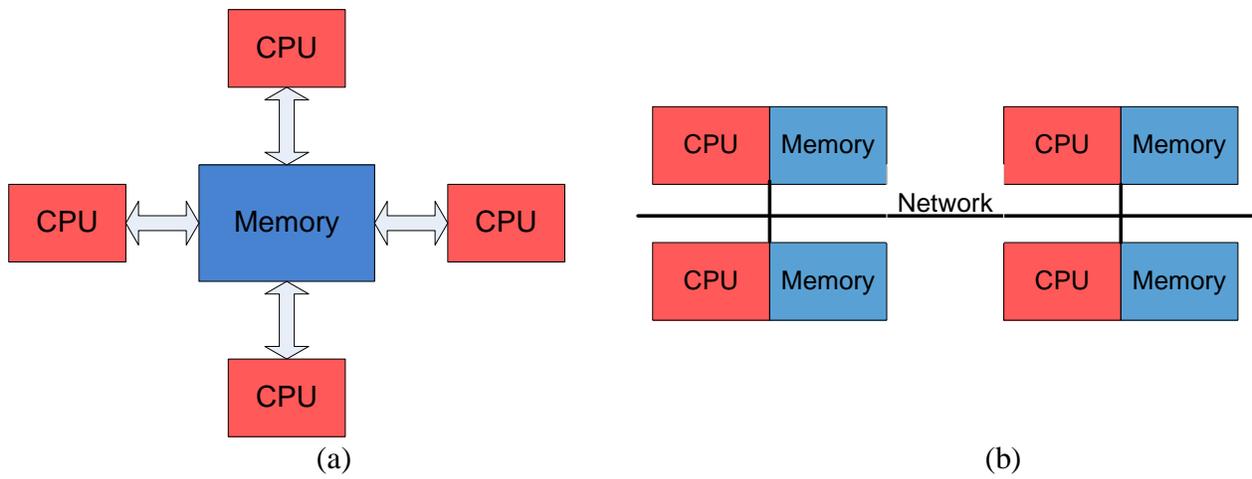


Figure 6-1. Memory architectures of the traditional parallel computers. (a) The shared memory architecture. (b) The distributed memory architecture.

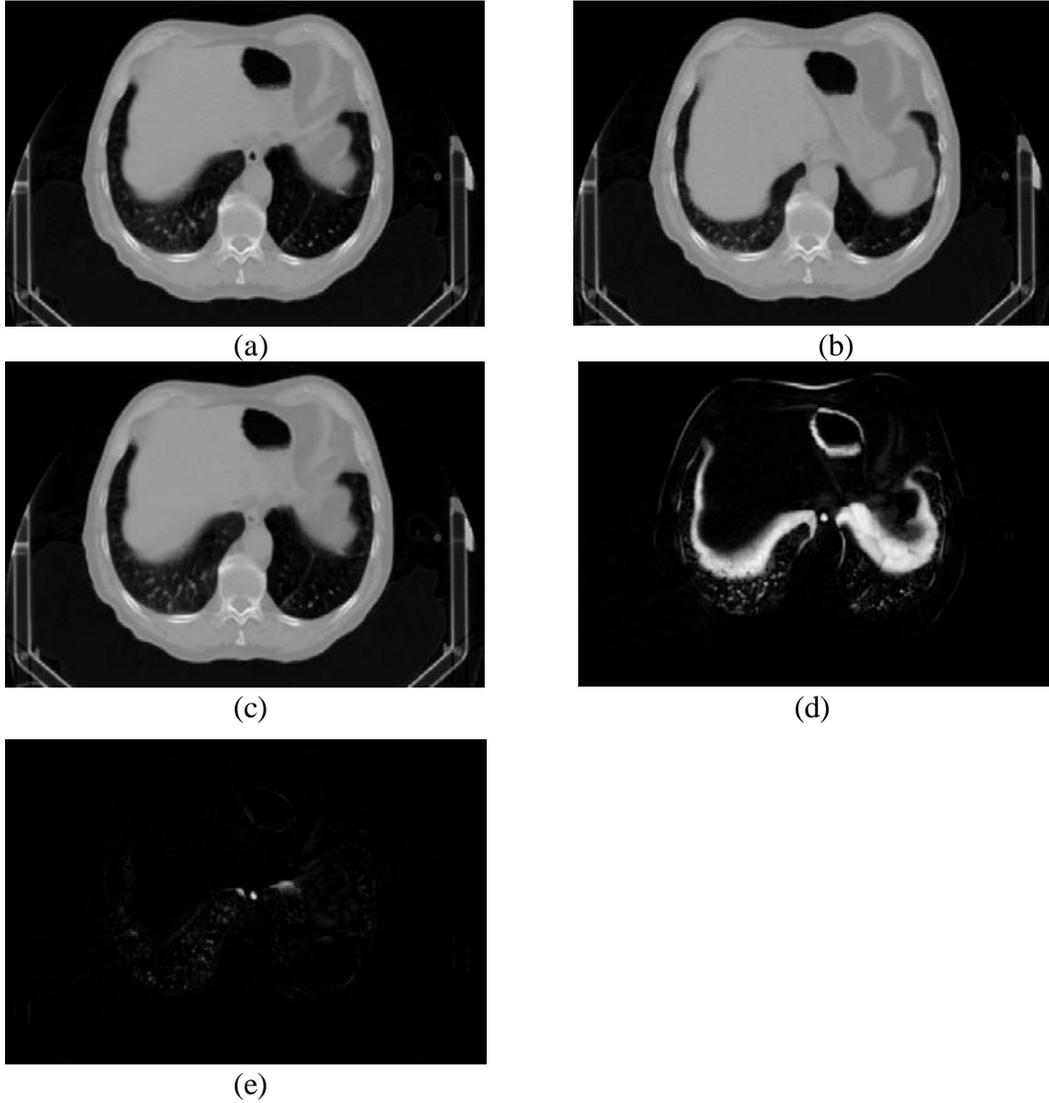


Figure 6-2. Deformable image registration using lung imaging. 2D image slices presented here are based on volumetric deformable image registration using demon's method. (a) Target image. (b) Source image. (c) Deformed image using demons deformable image registration on image (a). (d) Difference image of the source image and the target image. (e) Difference image of the target image and the deformed image (i.e. following DIR).

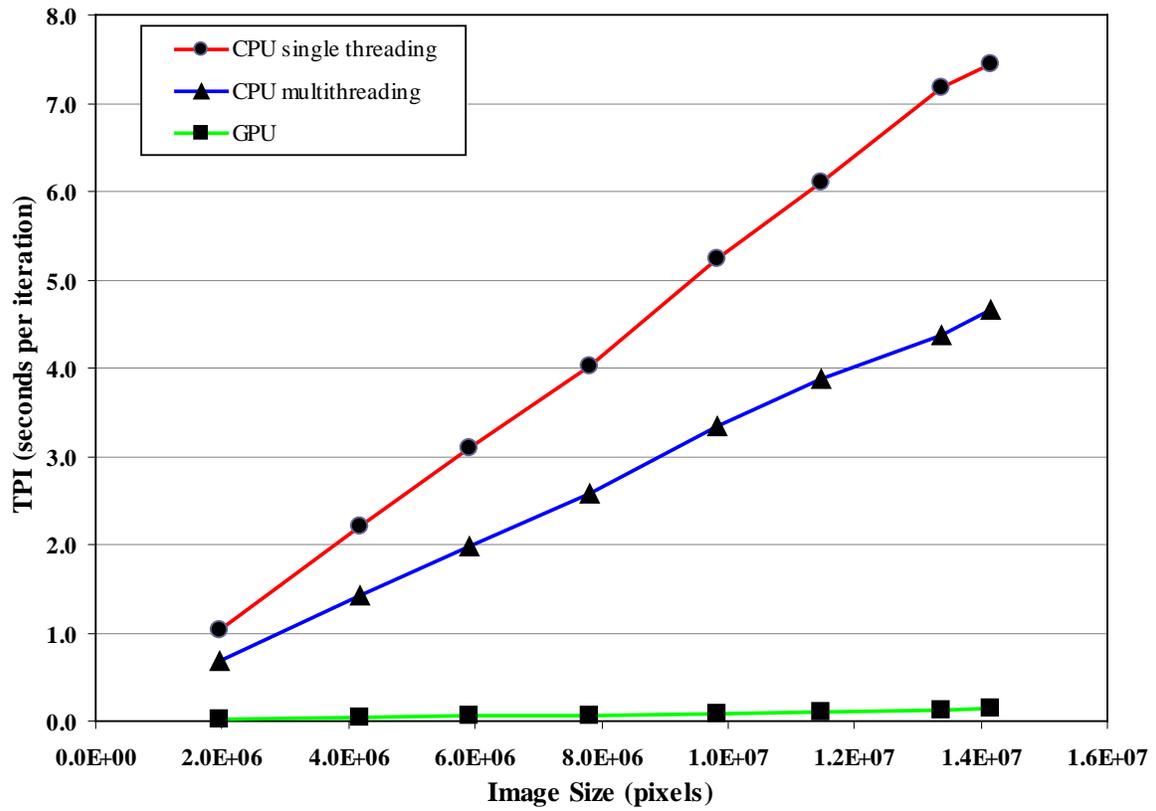


Figure 6-3. Performance comparisons between the CPU (single threading and multithreading) and the GPU based implementations of the Demons DIR algorithm as a function of image size. The GPU implementation is approximately 58 times faster than the corresponding single threading CPU implementation, and ~37 times faster than the corresponding multithreading CPU implementation. All three implementations exhibit a linear dependence on the image size.

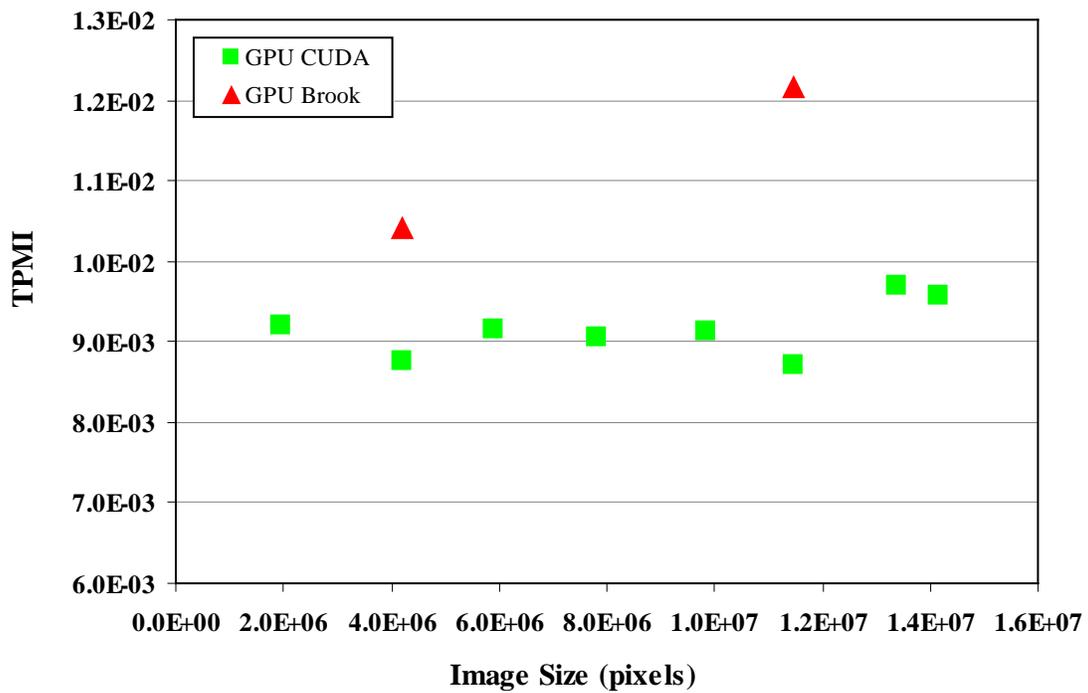


Figure 6-4. Computational efficiency of the GPU programming for DIR. The effect on data size on GPU computational efficiency was explored. Here we superimposed our GPU (NVIDIA 8800 GTX using CUDA programming language) results, alongside that of Sharp et al., who used NVIDIA 8800 GTS with the Brook programming language.

Table 6-1. Maximum, mean and standard deviation of the difference in correlation coefficients between the CPU implementations and the GPU implementation in 100 iterations

Patient	Max Difference	Mean Difference	Standard Derivation
1	2.2 E-5	3.0 E-6	4.0 E-6
2	2.1 E-5	4.0 E-6	4.0 E-6
3	3.0 E-5	7.0 E-6	8.0 E-6
4	3.1 E-5	7.0 E-6	8.0 E-6
5	3.0 E-5	5.1 E-6	5.0 E-6
6	2.5 E-5	3.7 E-6	5.0 E-6
7	2.6 E-5	4.0 E-6	4.2 E-6
8	2.3 E-5	3.3 E-6	4.0 E-6

Table 6-2. Performance comparison between the CPU and GPU based implementations of DIR for clinical imaging data

Patient	Image Size (pixels)	CPU Time (seconds)				GPU Time (seconds)	
		Single threading		Multithreading		TPI	TPMI
		TPI	TPMI	TPI	TPMI		
1	512x512x51	7.174	0.537	4.380	0.328	0.175	1.308E-2
2	512x512x54	7.453	0.526	4.660	0.329	0.186	1.316E-2
3	256x256x119	4.012	0.514	2.574	0.330	0.0713	9.138E-3
4	256x256x30	1.029	0.523	0.671	0.341	0.0181	9.21E-3
5	256x256x64	2.200	0.524	1.420	0.338	0.0367	8.75E-3
6	256x256x90	3.097	0.525	1.989	0.337	0.0540	9.16E-3
7	256x256x150	5.231	0.535	3.337	0.339	0.0898	9.13E-3
8	256x256x175	6.107	0.532	3.869	0.337	0.100	8.72E-3

Table 6-3. Performance comparison between the CPU (2.8GHz Intel Dual Core with 1.5GB RAM) and GPU (NVidia 8800 GTX) implementations of the Demons DIR algorithm for clinical imaging data in Sharp et al.'s GPU implementation

Data	Image Size	CPU time for single threading implementation (seconds)		GPU Time (seconds)	
		TPI	TPMI	TPI	TPMI
1	256x128x128	3.070	0.732	0.0437	0.0104
2	424x180x150	6.335	0.553	0.139	0.0122

## APPENDIX CODE SAMPLES OF RE-CONTOURING

This following re-contouring (propagateSingleTargetContoursByDIR.m) is to propagate 2D contours from the source image to the target image. The code is written using Matlab and python. The code was tested on Matlab 7.2 and VTK 5.2 with python 2.4 wrapper. To interact with Matlab, two additional python package, PyMat and SciPy, are also need to be installed. VTK python interface was used for surface reconstruction and post processing.

```
% ////////// propagateSingleTargetContoursByDIR.m //////////
% deform one single target contour by DIR
%
%
propagateSingleTargetContoursByDIR(deformationFields,dicomStruct,structureName)
%
% deformationfiels - 3D deformation fields
% dicomStruct - The contour structure exported as DICOM format
% structureName - name of the structure needed to deform. It is a
% cellarray, not caption sensitive. For example, {'heart'}.
%
% Junyi Xia
%
% Last Update: 05/03/08
% Revision: 0.1a
% //////////////////////////////////////
function vtkFileName =
propagateSingleTargetContoursByDIR(deformationFields,dicomStruct,structureName)
    vtkFileName = 'D:\test\DeformedSingleContour.vtk';
    PYTHON_BIN = 'C:\VTK\Binary\bin\release\vtkpython';

    CONVERTEDCONTOURFILENAME = 'D:\test\TempVTK.mat';
    SAVECONCONTOURFILENAME = 'D:\test\TempReconstructed.vtk';
    DEFORMEDCONTOURFILENAME = 'D:\test\TempVTK_Deformed.mat';

%% Step 1: reformat contours data for VTK usage

convertContour4Python2(dicomStruct,structureName,CONVERTEDCONTOURFILENAME);

%% Step 2: do VTK surface recontruction and save the result as vtk file
unixcommand = [PYTHON_BIN,' ','D:\Clinician version of
DRT\Segmentation\1.1\VTK\pyVTKSurfaceReconstruction.pyc',' ',
',CONVERTEDCONTOURFILENAME',' ',SAVECONCONTOURFILENAME];

    status = unix(unixcommand);
```

```

%% Step 3: Deforming the reconstructed surface points
% Read VTK points
singleStructure = readVTKContourPoints(SAVECONCONTOURFILENAME);

% Deforming points
deformedStructure = mappingContourPoints3D(deformationFields,
singleStructure);

% Save as .mat file for Python
save(DEFORMEDCONTOURFILENAME,'deformedStructure');
disp('Step 3 ... completed');

%% Step 4: Contour regeneration using planar cutting plane
% 1) Generate the new mesh with deformed contour points
% 2) Cut the new mesh with the parallel planes
unixcommand = [PYTHON_BIN, ' ', 'D:\Clinician version of
DRT\Segmentation\1.1\VTK\pyVTKCutDeformedSurface.pyc', '
',SAVECONCONTOURFILENAME, ' ',DEFORMEDCONTOURFILENAME, ' ',
num2str(floor(min(deformedStructure(:,3))))), '
',num2str(ceil(max(deformedStructure(:,3))))), ' ',vtkFileName];

status = unix(unixcommand);

```

```

% ////////////////////////////////// convertContour4Python2.m //////////////////////////////////
% convert the contours from vtk python surface reconstruction
%
%   converted =
convertContour4Python(structureFileName,fieldname,structureNameList,
saveFileName)
%
%   rtStructures - exported structure contour points.
%   structureNameList - all the interested structures, such as 'heart',
%   'cord'. It is a cell array.
%   saveFileName
%
% change logs:
%   1)
%
%   Junyi Xia
%   $ Revision: 0.1 $
%   $ Last update: 03/31/08 $
% //////////////////////////////////
function convertContour4Python2(rtStructures, structureNameList, saveFileName)

% Extract the points of the specified structure
selectedStructures = rtStructures;
for i=length(selectedStructures):-1:1
    strName = selectedStructures(i).Name;
    if ( ~isTargetStructure(structureNameList,strName))
        selectedStructures(i) = [];
    end
end
end

% Filtering small contours
selectedStructures = drcGetFilteredStructures(selectedStructures);

savedStructName = [];
for j = 1:length(selectedStructures)
    contourpoints = [];
    for i = 1:length(selectedStructures(j).Contours)
        temp = selectedStructures(j).Contours(i).Points;
        z = selectedStructures(j).Contours(i).Idx;
        temp = temp';
        temp(:,3) = z;
        contourpoints{i} = temp;
    end
    structName = strrep(selectedStructures(j).Name, '-', '_');
    savedStructName{j} = structName;
    eval(sprintf('%s = contourpoints;', structName));
end

save(saveFileName, savedStructName{:});
disp('Converting completed!')

function result = isTargetStructure( cellTargetStructure, input)
    n = length(cellTargetStructure);
    result = 0;

    for i=1:n

```

```
    if ( strcmpi(input,cellTargetStructure{i}))
        result = 1;
        break;
    end
end
```

pyVTKSurfaceReconstruction.py

*// Reconstruct the surface from 2D contours*

```
#!/usr/bin/env python
#
import vtk
import sys
from scipy.io import loadmat

# Create the RenderWindow, Renderer and both Actors
#
ren1 = vtk.vtkRenderer()
renWin = vtk.vtkRenderWindow()
renWin.AddRenderer(ren1)
iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renWin)

points = []
polys = []

#
# Load the data
#
loadpoints = loadmat(sys.argv[1])
tempContours = loadpoints.keys()

# Get contour names
contoursNames = []
for i in range(0,len(tempContours)):
    if ( tempContours[i].find('__') == -1 ):
        contoursNames.append(tempContours[i])

contours = []
contourMapper = []
contourActor = []
f = []
m = []
a = []

for k in range(0,len(contoursNames)):
    totalPoints = loadpoints[contoursNames[k]]
    numContours = len(totalPoints)

    idx = 0
    points.append(vtk.vtkPoints())
    polys.append(vtk.vtkCellArray())
    print contoursNames[k]
    for i in range(0,numContours):
        numPoints = len(totalPoints[i])
        print numPoints
        polys[k].InsertNextCell(numPoints)
        for j in range(0,numPoints):
```

```

points[k].InsertPoint(idx,totalPoints[i][j,0],totalPoints[i][j,1],totalPoints
[i][j,2])
    polys[k].InsertCellPoint(idx)
    idx += 1

#
# Create a representation of the contours used as input
#
contours.append(vtk.vtkPolyData())
contours[k].SetPoints(points[k])
contours[k].SetPolys(polys[k])

contourMapper.append(vtk.vtkPolyDataMapper())
contourMapper[k].SetInput(contours[k])

contourActor.append(vtk.vtkActor())
contourActor[k].SetMapper(contourMapper[k])
contourActor[k].GetProperty().SetColor(1,0,0)
contourActor[k].GetProperty().SetAmbient(1)
contourActor[k].GetProperty().SetDiffuse(0)
contourActor[k].GetProperty().SetRepresentationToWireframe()

ren1.AddActor(contourActor[k])

#
# Create the contour to surface filter
#
f.append(vtk.vtkVoxelContoursToSurfaceFilter())
f[k].SetInput(contours[k])
f[k].SetMemoryLimitInBytes(1000000)
f[k].SetSpacing(1,1,1)

m.append(vtk.vtkPolyDataMapper())
m[k].SetInputConnection(f[k].GetOutputPort())
m[k].ScalarVisibilityOff()
m[k].ImmediateModeRenderingOn()

a.append(vtk.vtkActor())
a[k].SetMapper(m[k])

ren1.AddActor(a[k])
contourActor[k].VisibilityOn()
#####
writer = vtk.vtkPolyDataWriter()
writer.SetFileName(sys.argv[2])
writer.SetInput(f[k].GetOutput())
writer.Write()
#####

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   writeVTKVolume2Matrix.m
%   read a vtk image file and write it as matlab matrix
%
%   Parameter:
%       vtkfile - name of the vtkfile
%   Return:
%       vol - image volume
%
%   Note: no spacing, origin information is returned.
%
%   Junyi Xia
%
%   Nov.30, 2006
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function points = readVTKContourPoints(vtkfile)

fid = fopen(vtkfile,'r','b');

if ( fid == -1 )
    error('can not open file %s !',vtkfile);
end

numberOfPoints = 0;
dims = [];
vol = [];
% read file line by line
while 1
    tline = fgetl(fid);
    % end of file?
    if ( tline == -1 )
        break;
    end
    % is a string ?
    if ( isempty(str2num(tline)))
        index = strfind(tline, 'POINTS');
        if ( ~isempty(index))
            numberOfPoints = str2num(sscanf(tline,'%*s %s %*s'));

            if ( numberOfPoints ~= 0 )
                temp = textscan(fid,'%f %f %f',numberOfPoints);
                points(:,1) = temp{1};
                points(:,2) = temp{2};
                points(:,3) = temp{3};
            end
            break;
        end
    end
end

vol = double(vol);
fclose(fid);

```

```

% ////////////////////////////////////////////////// mappingContourPoints3D.m //////////////////////////////////////////////////
% deform the contour points by the deformation fields.
%
%   mappedPoints = mappingContourPoints(deformationfields, contourpoints)
%
%   deformationfields - 3D deformation fields
%   contourpoints - contour points, it is a cell array
%
%   Junyi Xia
%
%   Last Update: 04/01/08
% //////////////////////////////////////////////////
function mappedContourPoints = mappingContourPoints3D(deformationfields,
contourpoints)

% rebuild the points into n x 3 matrix
contour = contourpoints;
mappedPoints = [];

% do the interpolation
mappedPoints(:,1) =
interp3(deformationfields(:,:,1),contour(:,1),contour(:,2),contour(:,3),'li
near',0);
mappedPoints(:,2) =
interp3(deformationfields(:,:,2),contour(:,1),contour(:,2),contour(:,3),'li
near',0);
mappedPoints(:,3) =
interp3(deformationfields(:,:,3),contour(:,1),contour(:,2),contour(:,3),'li
near',0);

mappedContourPoints = mappedPoints + contour;

```

pyVTKCutDeformedSurface.py

*// Generate 2D deformed contours from cutting 3D deformed surface mesh*

```
#!/usr/bin/env python
#
## argv[1] - reconstructed mesh
## argv[2] - deformed points
## argv[3] - minimum z of cutting plane
## argv[4] - maximum z of cutting planeimport vtk

import vtk
import sys
from scipy.io import loadmat

# Load the deformed points
loadpoints = loadmat(sys.argv[2])
contourPoints = loadpoints['deformedStructure']

# Load the original mesh
fileReader = vtk.vtkPolyDataReader()
fileReader.SetFileName(sys.argv[1])
fileReader.Update()
polyMesh = fileReader.GetOutput()

ptsDeformed = vtk.vtkPoints()

# Build points with deformed contour points
idx = 0
for i in range(0,len(contourPoints)):

ptsDeformed.InsertPoint(idx,contourPoints[i][0],contourPoints[i][1],contourPo
ints[i][2])
    idx += 1

polyMesh.SetPoints(ptsDeformed)

# Get the max and min Z of the cutting plane
minZ = int(sys.argv[3])
maxZ = int(sys.argv[4])

plane = []
planeCut = []
appendContours = vtk.vtkAppendPolyData()
for i in range(0,maxZ-minZ+1):
    plane.append(vtk.vtkPlane())
    if i == 0:
        plane[i].SetOrigin(0,0,minZ+0.00001)
    elif i == maxZ-minZ:
        plane[i].SetOrigin(0,0,maxZ-0.00001)
    else:
        plane[i].SetOrigin(0,0,minZ+i+0.00001)
```

```
plane[i].SetNormal(0,0,1)
planeCut.append(vtk.vtkCutter())
planeCut[i].SetCutFunction(plane[i])
planeCut[i].SetInput(polyMesh)
appendContours.AddInput(planeCut[i].GetOutput())

writer2 = vtk.vtkPolyDataWriter()
writer2.SetFileName(sys.argv[5])
writer2.SetInput(appendContours.GetOutput())
writer2.Update()
writer2.Write()
```

## LIST OF REFERENCES

1. J. V. Dyk, *The Modern Technology of Radiation Oncology: A Compendium for Medical Physicists and Radiation Oncologists* (Medical Physics Publishing Corporation, Madison, Wisconsin, 1999).
2. J. M. Balter, H. M. Sandler, K. Lam, R. L. Bree, A. S. Lichter and R. K. ten Haken, "Measurement of prostate movement over the course of routine radiotherapy using implanted markers," *Int J Radiat Oncol Biol Phys* **31**, 113-118 (1995).
3. C. S. Ross, D. H. Hussey, E. C. Pennington, W. Stanford and J. F. Doornbos, "Analysis of movement of intrathoracic neoplasms using ultrafast computerized tomography," *Int J Radiat Oncol Biol Phys* **18**, 671-677 (1990).
4. K. M. Langen and D. T. Jones, "Organ motion and its management," *Int J Radiat Oncol Biol Phys* **50**, 265-278 (2001).
5. J. H. Heinzerling, J. F. Anderson, L. Papiez, T. Boike, S. Chien, G. Zhang, R. Abdulrahman and R. Timmerman, "Four-dimensional computed tomography scan analysis of tumor and organ motion at varying levels of abdominal compression during stereotactic treatment of lung and liver," *Int J Radiat Oncol Biol Phys* **70**, 1571-1578 (2008).
6. M. G. Herman, T. M. Pisansky, J. J. Kruse, J. I. Prisciandaro, B. J. Davis and B. F. King, "Technical aspects of daily online positioning of the prostate for three-dimensional conformal radiotherapy using an electronic portal imaging device," *Int J Radiat Oncol Biol Phys* **57**, 1131-1140 (2003).
7. S. Aubin, L. Beaulieu, S. Pouliot, J. Pouliot, R. Roy, L. M. Girouard, N. Martel-Brisson, E. Vigneault and J. Laverdiere, "Robustness and precision of an automatic marker detection algorithm for online prostate daily targeting using a standard V-EPID," *Med Phys* **30**, 1825-1832 (2003).
8. E. Vigneault, J. Pouliot, J. Laverdiere, J. Roy and M. Dorion, "Electronic portal imaging device detection of radioopaque markers for the evaluation of prostate position during megavoltage irradiation: a clinical study," *Int J Radiat Oncol Biol Phys* **37**, 205-212 (1997).
9. P. B. Greer, C. C. Jose and J. H. Matthews, "Set-up variation of patients treated with radiotherapy to the prostate measured with an electronic portal imaging device," *Australas Radiol* **42**, 207-212 (1998).
10. J. A. Stryker, J. Shafer and R. E. Beatty, "Assessment of accuracy of daily set-ups in prostate radiotherapy using electronic imaging," *Br J Radiol* **72**, 579-583 (1999).
11. S. L. Meeks, W. A. Tome, T. R. Willoughby, P. A. Kupelian, T. H. Wagner, J. M. Buatti and F. J. Bova, "Optically guided patient positioning techniques," *Semin Radiat Oncol* **15**, 192-201 (2005).

12. D. P. Gierga, J. Brewer, G. C. Sharp, M. Betke, C. G. Willett and G. T. Chen, "The correlation between internal and external markers for abdominal tumors: implications for respiratory gating," *Int J Radiat Oncol Biol Phys* **61**, 1551-1558 (2005).
13. P. Kupelian, T. Willoughby, A. Mahadevan, T. Djemil, G. Weinstein, S. Jani, C. Enke, T. Solberg, N. Flores, D. Liu, D. Beyer and L. Levine, "Multi-institutional clinical experience with the Calypso System in localization and continuous, real-time monitoring of the prostate gland during external radiotherapy," *Int J Radiat Oncol Biol Phys* **67**, 1088-1098 (2007).
14. T. R. Willoughby, P. A. Kupelian, J. Pouliot, K. Shinohara, M. Aubin, M. Roach, 3rd, L. L. Skrumeda, J. M. Balter, D. W. Litzenberg, S. W. Hadley, J. T. Wei and H. M. Sandler, "Target localization and real-time tracking using the Calypso 4D localization system in patients with localized prostate cancer," *Int J Radiat Oncol Biol Phys* **65**, 528-534 (2006).
15. D. Letourneau, J. W. Wong, M. Oldham, M. Gulam, L. Watt, D. A. Jaffray, J. H. Siewerdsen and A. A. Martinez, "Cone-beam-CT guided radiation therapy: technical implementation," *Radiother Oncol* **75**, 279-286 (2005).
16. G. G. Zeng, S. L. Breen, A. Bayley, E. White, H. Keller, L. Dawson and D. A. Jaffray, "A method to analyze the cord geometrical uncertainties during head and neck radiation therapy using cone beam CT," *Radiother Oncol* (2008).
17. F. Xu, J. Wang, S. Bai, Y. Li, Y. Shen, R. Zhong, X. Jiang and Q. Xu, "Detection of intrafractional tumour position error in radiotherapy utilizing cone beam computed tomography," *Radiother Oncol* (2008).
18. J. A. Tanyi and M. H. Fuss, "Volumetric image-guidance: does routine usage prompt adaptive re-planning? An institutional review," *Acta Oncol* **47**, 1444-1453 (2008).
19. L. Masi, F. Casamassima, C. Polli, C. Menichelli, I. Bonucci and C. Cavedon, "Cone beam CT image guidance for intracranial stereotactic treatments: comparison with a frame guided set-up," *Int J Radiat Oncol Biol Phys* **71**, 926-933 (2008).
20. M. M. Coselmon, J. M. Balter, D. L. McShan and M. L. Kessler, "Mutual information based CT registration of the lung at exhale and inhale breathing states using thin-plate splines," *Med Phys* **31**, 2942-2948 (2004).
21. H. H. Liu, P. Balter, T. Tutt, B. Choi, J. Zhang, C. Wang, M. Chi, D. Luo, T. Pan, S. Hunjan, G. Starkschall, I. Rosen, K. Prado, Z. Liao, J. Chang, R. Komaki, J. D. Cox, R. Mohan and L. Dong, "Assessing respiration-induced tumor motion and internal target volume using four-dimensional computed tomography for radiotherapy of lung cancer," *Int J Radiat Oncol Biol Phys* **68**, 531-540 (2007).
22. H. Wang, L. Dong, J. O'Daniel, R. Mohan, A. S. Garden, K. K. Ang, D. A. Kuban, M. Bonnen, J. Y. Chang and R. Cheung, "Validation of an accelerated 'demons' algorithm for deformable image registration in radiation therapy," *Phys Med Biol* **50**, 2887-2905 (2005).

23. G. E. Christensen, S. Joo Hyun, L. Wei, I. E. Naqa and D. A. Low, "Tracking lung tissue motion and expansion/compression with inverse consistent image registration and spirometry," *Medical Physics* **34**, 2155-2163 (2007).
24. K. Kitamura, H. Shirato, Y. Seppenwoolde, T. Shimizu, Y. Kodama, H. Endo, R. Onimaru, M. Oda, K. Fujita, S. Shimizu and K. Miyasaka, "Tumor location, cirrhosis, and surgical history contribute to tumor movement in the liver, as measured during stereotactic irradiation using a real-time tumor-tracking radiotherapy system," *Int J Radiat Oncol Biol Phys* **56**, 221-228 (2003).
25. K. K. Brock, M. B. Sharpe, L. A. Dawson, S. M. Kim and D. A. Jaffray, "Accuracy of finite element model-based multi-organ deformable image registration," *Med Phys* **32**, 1647-1659 (2005).
26. K. K. Brock, L. A. Dawson, M. B. Sharpe, D. J. Moseley and D. A. Jaffray, "Feasibility of a novel deformable image registration technique to facilitate classification, targeting, and monitoring of tumor and normal tissue," *Int J Radiat Oncol Biol Phys* (2006).
27. T. Rohlfing, C. R. Maurer, Jr., W. G. O'Dell and J. Zhong, "Modeling liver motion and deformation during the respiratory cycle using intensity-based nonrigid registration of gated MR images," *Med Phys* **31**, 427-432 (2004).
28. W. Lu, G. H. Olivera, Q. Chen, M.-L. Chen and K. J. Ruchala, "Automatic re-contouring in 4D radiotherapy," *Physics in Medicine and Biology* **51**, 1077-1099 (2006).
29. D. L. Collins, A. C. Evans, C. Holmes and T. M. Peters, "Automatic 3D segmentation of neuro-anatomical structures from MRI," in *information processing in medical imaging: Proc. 14th international conference*, edited by B. Y., C. and di Paola, R. (Kluwer Academic Publishers, Dordrecht, 1995), pp. 139-152.
30. J.-P. Thirion, "Image matching as a diffusion process: an analogy with Maxwell's demons," *Medical Image Analysis* **2**, 243-260 (1998).
31. W. Lu, M. L. Chen, G. H. Olivera, K. J. Ruchala and T. R. Mackie, "Fast free-form deformable registration via calculus of variations," *Phys Med Biol* **49**, 3067-3087 (2004).
32. B. Fischer and J. Modersitzki, "Fast diffusion registration," *AMS contemporary Mathematics, Inverse Problems, Image Analysis, and Medical Imaging* **313**, 117-129 (2002).
33. G. E. Christensen, "Minimizing sources of errors in medical image registration," (IEEE, Washington, DC, USA, 2002), pp. 14-17.
34. D. Sarrut, "Deformable registration for image-guided radiation therapy," *Z Med Phys* **16**, 285-297 (2006).
35. B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing* **21**, 977-1000 (2003).

36. J. B. Maintz and M. A. Viergever, "A survey of medical image registration," *Med Image Anal* **2**, 1-36 (1998).
37. H. Lester and S. R. Arridge, "A survey of hierarchical non-linear medical image registration," *Pattern Recognition* **32**, 129-149 (1999).
38. D. L. Hill, P. G. Batchelor, M. Holden and D. J. Hawkes, "Medical image registration," *Phys Med Biol* **46**, R1-45 (2001).
39. A. Gholipour, N. Kehtarnavaz, R. Briggs, M. Devous and K. Gopinath, "Brain functional localization: a survey of image registration techniques," *IEEE Trans Med Imaging* **26**, 427-451 (2007).
40. B. M. Dawant, "Non-rigid registration of medical images: purpose and methods, a short survey," (IEEE, Washington, DC, USA, 2002), pp. 465-468.
41. L. Brown, "A survey of image registration techniques," *Computing Surveys* **24**, 325-376 (1992).
42. J. P. W. Pluim, J. B. A. Maintz and M. A. Viergever, "Mutual-information-based registration of medical images: a survey," *IEEE TRANSACTIONS ON MEDICAL IMAGING* **22**, 986-1004 (2003).
43. W. R. Crum, T. Hartkens and D. L. Hill, "Non-rigid image registration: theory and practice," *Br J Radiol* **77 Spec No 2**, S140-153 (2004).
44. D. J. Hawkes, D. Barratt, J. M. Blackall, C. Chan, P. J. Edwards, K. Rhode, G. P. Penney, J. McClelland and D. L. Hill, "Tissue deformation and shape models in image-guided interventions: a discussion paper," *Med Image Anal* **9**, 163-175 (2005).
45. T. T. McInerney, D, "Deformable Models in medical Image Analysis: a Survey," *Medical Image Analysis* **1**, 91-108 (1996).
46. J. V. Hajnal, D. J. Hawkes and D. Hill, *Medical image registration*. (CRC Press, Boca Raton, 2001).
47. C. Zhujiang, P. Shiyan, L. Rui, R. Balachandran, J. M. Fitzpatrick and W. C. Chapman, "Registration of medical images using an interpolated closest point transform: method and validation," *Medical Image Analysis* **8**, 421-427 (2004).
48. J. B. West and J. M. Fitzpatrick, "Point-based rigid registration: clinical validation of theory," *Vol. 3979*, (SPIE-Int. Soc. Opt. Eng, San Diego, CA, USA, 2000), pp. 353-359.
49. K. Rohr, H. S. Stiehl, R. Sprengel, W. Beil, T. M. Buzug, J. Weese and M. H. Kuhn, "Point-based elastic registration of medical image data using approximating thin-plate splines," (Springer-Verlag, Hamburg, Germany, 1996), pp. 297-306.

50. K. Rohr, *Landmark-based image analysis : using geometric and intensity models*. (Kluwer Academic Publishers, Dordrecht ; Boston, 2001).
51. H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding* **89**, 114-141 (2003).
52. W. Beil, K. Rohr and H. S. Stiehl, "Investigation of approaches for the localization of anatomical landmarks in 3D medical images," (Elsevier, Berlin, Germany, 1998), pp. 265-270.
53. P. J. Besl and H. D. McKay, "A method for registration of 3-D shapes," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* **14**, 239-256 (1992).
54. T. Masuda and N. Yokoya, "Robust method for registration and segmentation of multiple range images," *Computer Vision and Image Understanding* **61**, 295-307 (1995).
55. T. Zinsser, J. Schmidt and H. Niemann, "A refined ICP algorithm for robust 3-D correspondence estimation," *Vol. 2*, (Institute of Electrical and Electronics Engineers Computer Society, Barcelona, Spain, 2003), pp. 695-698.
56. M. Rogers and J. Graham, "Robust and accurate registration of 2-D electrophoresis gels using point-matching," *IEEE Transactions on Image Processing* **16**, 624-635 (2007).
57. M. H. Mahoor and M. Abdel-Mottaleb, "Face recognition based on 3D ridge images obtained from range data," *Pattern Recognition* **42**, 445-451 (2009).
58. S. Sclaroff and A. P. Pentland, "Modal matching for correspondence and recognition," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* **17**, 545-561 (1995).
59. A. D. J. Cross and E. R. Hancock, "Graph matching with a dual-step EM algorithm," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* **20**, 1236-1253 (1998).
60. G. Subsol, N. Roberts, M. Doran, J. P. Thirion and G. H. Whitehouse, "Automatic analysis of cerebral atrophy," *Magnetic Resonance Imaging* **15**, 917-927 (1997).
61. J. B. A. Maintz, P. A. van den Elsen and M. A. Viergever, "Evaluation of ridge seeking operators for multimodality medical image matching," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* **18**, 353-365 (1996).
62. P. A. Van den Elsen, J. B. A. Maintz, E.-J. D. Pol and M. A. Viergever, "Automatic registration of CT and MR brain images using correlation of geometrical features," *IEEE TRANSACTIONS ON MEDICAL IMAGING* **14**, 384-396 (1995).

63. O. Monga, S. Benayoun and O. D. Faugeras, "From partial derivatives of 3-D density images to ridge lines," (IEEE Comput. Soc. Press, Champaign, IL, USA, 1992), pp. 354-359.
64. A. Gueziec and N. Ayache, "Smoothing and matching of 3-D space curves," (Springer-Verlag, Santa Margherita Ligure, Italy, 1992), pp. 620-629.
65. B. K. P. Horn and B. G. Schunck, "DETERMINING OPTICAL FLOW," *Artificial Intelligence* **17**, 185-203 (1981).
66. G. E. Christensen and H. J. Johnson, "Consistent image registration," *IEEE Trans Med Imaging* **20**, 568-582 (2001).
67. A. Roche, G. Malandain, N. Ayache and S. Prima, "Towards a better comprehension of similarity measures used in medical image registration," edited by C. C. A. Taylor (Springer-Verlag Berlin, 1999), pp. 555-566.
68. R. Shekhar and V. Zagrodsky, "Mutual information-based rigid and nonrigid registration of ultrasound volumes," *IEEE Trans Med Imaging* **21**, 9-22 (2002).
69. Z. Zhang, Y. Jiang and H. Tsui, "Consistent multi-modal non-rigid registration based on a variational approach," *Pattern Recognition Letters* **27**, 715-725 (2006).
70. A. Andronache, M. von Siebenthal, G. Szekely and P. Cattin, "Non-rigid registration of multi-modal images using both mutual information and cross-correlation," *Medical Image Analysis* **12**, 3-15 (2008).
71. G. E. Christensen, B. Carlson, K. S. Chao, P. Yin, P. W. Grigsby, K. Nguyen, J. F. Dempsey, F. A. Lerma, K. T. Bae, M. W. Vannier and J. F. Williamson, "Image-based dose planning of intracavitary brachytherapy: registration of serial-imaging studies using deformable anatomic templates," *Int J Radiat Oncol Biol Phys* **51**, 227-243 (2001).
72. J. P. Pluim, J. B. Maintz and M. A. Viergever, "Image registration by maximization of combined mutual information and gradient information," *IEEE Trans Med Imaging* **19**, 809-814 (2000).
73. P. Hellier and C. Barillot, "Coupling dense and landmark-based approaches for nonrigid registration," *IEEE Trans Med Imaging* **22**, 217-227 (2003).
74. P. Cachier, E. Bardinet, D. Dormont, X. Pennec and N. Ayache, "Iconic feature based nonrigid registration: the PASHA algorithm," *Computer Vision and Image Understanding* **89**, 272-298 (2003).
75. D. L. Collins, P. Neelin, T. M. Peters and A. C. Evans, "Automatic 3D intersubject registration of MR volumetric data in standardized Talairach space," *J Comput Assist Tomogr* **18**, 192-205 (1994).

76. X. Wu, S. J. Dibiase, R. Gullapalli and C. X. Yu, "Deformable image registration for the use of magnetic resonance spectroscopy in prostate treatment planning," *Int J Radiat Oncol Biol Phys* **58**, 1577-1583 (2004).
77. R. L. Harder and R. N. Desmarais, "Interpolation using surface splines," *Journal of Aircraft* **9**, 189-191 (1972).
78. A. Goshtasby, "Registration of image with geometric distortion," *IEEE Transactions of Geoscience and Remote Sensing* **26**, 60-64 (1988).
79. J. Lian, L. Xing, S. Hunjan, C. Dumoulin, J. Levin, A. Lo, R. Watkins, K. Rohling, R. Giaquinto, D. Kim, D. Spielman and B. Daniel, "Mapping of the prostate in endorectal coil-based MRI/MRSI and CT: a deformable registration and validation study," *Med Phys* **31**, 3087-3094 (2004).
80. H. J. Johnson and G. E. Christensen, "Consistent landmark and intensity-based image registration," *Vol. 21*, (IEEE, Davis, CA, USA, 2002), pp. 450-461.
81. J. E. Downhill, Jr., M. S. Buchsbaum, T. Wei, J. Spiegel-Cohen, E. A. Hazlett, M. M. Haznedar, J. Silverman and L. J. Siever, "Shape and size of the corpus callosum in schizophrenia and schizotypal personality disorder," *Schizophr Res* **42**, 193-208 (2000).
82. L. Zagorchev and A. Goshtasby, "A comparative study of transformation functions for nonrigid image registration," *Image Processing, IEEE Transactions on* **15**, 529-538 (2006).
83. D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach and D. J. Hawkes, "Nonrigid registration using free-form deformations: application to breast MR images," *IEEE Trans Med Imaging* **18**, 712-721 (1999).
84. Z. Xie and G. E. Farin, "Image registration using hierarchical B-splines," *IEEE Trans Vis Comput Graph* **10**, 85-94 (2004).
85. D. Mattes, D. R. Haynor, H. Vesselle, T. K. Lewellen and W. Eubank, "PET-CT image registration in the chest using free-form deformations," *IEEE Trans Med Imaging* **22**, 120-128 (2003).
86. A. F. Frangi, D. Rueckert, J. A. Schnabel and W. J. Niessen, "Automatic construction of multiple-object three-dimensional statistical shape models: application to cardiac modeling," *IEEE Trans Med Imaging* **21**, 1151-1166 (2002).
87. K. McLeish, D. L. Hill, D. Atkinson, J. M. Blackall and R. Razavi, "A study of the motion and deformation of the heart due to respiration," *IEEE Trans Med Imaging* **21**, 1142-1150 (2002).
88. M. Ferrant, A. Nabavi, B. Macq, F. A. Jolesz, R. Kikinis and S. K. Warfield, "Registration of 3-D intraoperative MR images of the brain using a finite-element biomechanical model," *IEEE Trans Med Imaging* **20**, 1384-1397 (2001).

89. A. Hagemann, K. Rohr, H. S. Stiehl, U. Spetzger and J. M. Gilsbach, "Biomechanical Modeling of the Human Head for Physically Based, Nonrigid Image Registration," *IEEE TRANSACTIONS ON MEDICAL IMAGING* **18**, 875-884 (1999).
90. A. Mohamed, E. I. Zacharaki, D. Shen and C. Davatzikos, "Deformable registration of brain tumor images via a statistical model of tumor-induced deformation," *Med Image Anal* (2006).
91. A. Samani, J. Bishop, M. J. Yaffe and D. B. Plewes, "Biomechanical 3-D finite element modeling of the human breast using MRI data," *IEEE Trans Med Imaging* **20**, 271-279 (2001).
92. J. A. Schnabel, C. Tanner, A. D. Castellano-Smith, A. Degenhard, M. O. Leach, D. R. Hose, D. L. G. Hill and D. J. Hawkes, "Validation of nonrigid image registration using finite-element methods: application to breast MR images," *IEEE Transactions on Medical Imaging* **22**, 238 - 247 (2003).
93. M. Sermesant, P. Moireau, O. Camara, J. Sainte-Marie, R. Andriantsimiavona, R. Cimrman, D. L. Hill, D. Chapelle and R. Razavi, "Cardiac function estimation from MRI using a heart model and data assimilation: advances and difficulties," *Med Image Anal* **10**, 642-656 (2006).
94. K. K. Brock, D. L. McShan, R. K. Ten Haken, S. J. Hollister, L. A. Dawson and J. M. Balter, "Inclusion of organ deformation in dose calculations," *Med Phys* **30**, 290-295 (2003).
95. T. Zhang, N. P. Orton, T. R. Mackie and B. R. Paliwal, "Technical note: A novel boundary condition using contact elements for finite element based deformable image registration," *Med Phys* **31**, 2412-2415 (2004).
96. Y. Chi, J. Liang and D. Yan, "A material sensitivity study on the accuracy of deformable organ registration using linear biomechanical models," *Med Phys* **33**, 421-433 (2006).
97. P. Hellier, C. Barillot, E. Memin and P. Perez, "Hierarchical estimation of a dense deformation field for 3-D robust registration," *IEEE Trans Med Imaging* **20**, 388-402 (2001).
98. J. C. Gee, M. Reivich and R. Bajcsy, "Elastically deforming 3D atlas to match anatomical brain images," *J Comput Assist Tomogr* **17**, 225-236 (1993).
99. R. Bajcsy and S. Kovacic, "Multiresolution Elastic Matching," *Computer Vision, Graphics, and Image Processing* **46**, 1-21 (1989).
100. M. Bro-Nielsen and C. Gramkow, "Fast Fluid Registration of Medical Images," *Visualization in Biomedical Computing, Lecture Notes in Computer Science* **1131**, 267-276 (1996).

101. E. D'Agostino, F. Maes, D. Vandermeulen and P. Suetens, "A viscous fluid model for multimodal non-rigid image registration using mutual information," *Med Image Anal* **7**, 565-575 (2003).
102. P. A. Freeborough and N. C. Fox, "Modeling brain deformations in Alzheimer disease by fluid registration of serial 3D MR images," *Journal of Computer Assisted Tomography* **22**, 838-843 (1998).
103. F. L. Bookstein, "Thin-plate splines and the decomposition of deformations," *Lect. Notes Comput. Sci.* **511**, 326-342 (1991).
104. D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8**, 413-424 (1986).
105. W. H. Press, *Numerical recipes : the art of scientific computing*. (Cambridge University Press, Cambridge ; New York, 1989).
106. N. Paragios, Y. Chen and O. Faugeras, *Handbook of mathematical models in computer vision*. (Springer, New York, 2006).
107. M. Chen, W. Lu, Q. Chen, K. J. Ruchala and G. H. Olivera, "A simple fixed-point approach to invert a deformation field," *Med Phys* **35**, 81-88 (2008).
108. P. Cachier and D. Rey, "Symmetrization of the non-rigid registration problem using inversion-invariant energies: application to multiple sclerosis," (Springer-Verlag, Pittsburgh, PA, USA, 2000), pp. 472-481.
109. J. Ashburner, J. L. Andersson and K. J. Friston, "Image registration using a symmetric prior--in three dimensions," *Hum Brain Mapp* **9**, 212-225 (2000).
110. T. S. Yoo, *Insight into Images: Principles and Practice for Segmentation, Registration, and Image Analysis*. (AK Peters, Ltd, 2004).
111. I. Arganda-Carreras, C. O. S. Sorzano, R. Marabini, J. M. Carazo, C. Ortiz-de-Solorzano and J. Kybic, "Consistent and elastic registration of histological sections using vector-spline regularization," (Springer-Verlag, Graz, Austria, 2006), pp. 85-95.
112. V. A. Magnotta, H. J. Bockholt, H. J. Johnson, G. E. Christensen and N. C. Andreasen, "Subcortical, cerebellar, and magnetic resonance based consistent brain image registration," *Neuroimage* **19**, 233-245 (2003).
113. J. He and G. E. Christensen, "Large deformation inverse consistent elastic image registration," *Inf Process Med Imaging* **18**, 438-449 (2003).
114. G. E. Christensen, "Inverse consistent registration with object boundary constraints," *Vol. I*, (IEEE, Arlington, VA, USA, 2004), pp. 591-594.

115. S. K. Yeung and P. Shi, "Stochastic inverse consistent registration," (IEEE, Shanghai, China, 2006), pp. 4 pp.
116. S. K. Yeung and P. C. Shi, "Stochastic inverse consistency in medical image registration," in *Medical Image Computing and Computer-Assisted Intervention - Miccai 2005, Pt 2, Vol. 3750*, (Springer-Verlag Berlin, Berlin, 2005), pp. 188-196.
117. L. Alvarez, R. Deriche, T. Papadopoulo and J. Sanchez, "Symmetrical dense optical flow estimation with occlusions detection," (Springer-Verlag, Copenhagen, Denmark, 2002), pp. 721-735.
118. A. Leow, S.-C. Huang, A. Geng, J. Becker, S. Davis, A. Toga and P. Thompson, "Inverse consistent mapping in 3D deformable image registration: Its construction and statistical properties," *Vol. 3565*, (Springer Verlag, Heidelberg, D-69121, Germany, Glenwood Springs, CO, United States, 2005), pp. 493-503.
119. B. B. Avants, M. Grossman and J. C. Gee, "Symmetric diffeomorphic image registration: Evaluating automated labeling of elderly and neurodegenerative cortex and frontal lobe," *Vol. 4057 LNCS*, (Springer Verlag, Heidelberg, D-69121, Germany, Utrecht, Netherlands, 2006), pp. 50-57.
120. M. F. Beg and A. Khan, "Symmetric data attachment terms for large deformation image registration," *IEEE TRANSACTIONS ON MEDICAL IMAGING* **26**, 1179-1189 (2007).
121. S. Joshi, B. Davis, M. Jomier and G. Gerig, "Unbiased diffeomorphic atlas construction for computational anatomy," *Neuroimage* **23 Suppl 1**, S151-160 (2004).
122. G. Xiujuan, D. Kumar, G. E. Christensen and M. W. Vannier, "Inverse consistent image registration of MR brain scans: handedness in normal adult males," (Springer-Verlag, Philadelphia, PA, USA, 2003), pp. 71-80.
123. P. Jannin, J. M. Fitzpatrick, D. J. Hawkes, X. Pennec, R. Shahidi and M. W. Vannier, "Validation of medical image processing in image-guided therapy," *IEEE Trans Med Imaging* **21**, 1445-1449 (2002).
124. R. Kashani, M. Hub, M. L. Kessler and J. M. Balter, "Technical note: a physical phantom for assessment of accuracy of deformable alignment algorithms," *Med Phys* **34**, 2785-2788 (2007).
125. M. Serban, E. Heath, G. Stroian, D. L. Collins and J. Seuntjens, "A deformable phantom for 4D radiotherapy verification: design and image registration evaluation," *Med Phys* **35**, 1094-1102 (2008).
126. J. G. Parker and D. R. Gilland, "Wall motion estimation for gated cardiac emission tomography: Physical phantom evaluation," *IEEE Transactions on Nuclear Science* **55**, 531-536 (2008).

127. A. Gholipour, N. Kehtarnavaz, R. W. Briggs, K. S. Gopinath, W. Ringe, A. Whittemore, S. Cheshkov and K. Bakhadirov, "Validation of Non-Rigid Registration Between Functional and Anatomical Magnetic Resonance Brain Images," *Biomedical Engineering, IEEE Transactions on* **55**, 563-571 (2008).
128. Z. Xue, D. Shen, B. Karacali, J. Stern, D. Rottenberg and C. Davatzikos, "Simulating deformations of MR brain images for validation of atlas-based segmentation and registration algorithms," *Neuroimage* **33**, 855-866 (2006).
129. R. Fahmi, A. Aly, A. Elbaz and A. A. Farag, "New deformable registration technique using scale space and curve evolution theory and a finite element based validation framework," *Conf Proc IEEE Eng Med Biol Soc* **1**, 3041-3044 (2006).
130. P. Castadot, J. A. Lee, A. Parraga, X. Geets, B. Macq and V. Gregoire, "Comparison of 12 deformable registration strategies in adaptive radiation therapy for the treatment of head and neck tumors," *Radiother Oncol* (2008).
131. G. E. Christensen, X. Geng, J. G. Kuhl, J. Bruss, T. J. Grabowski, I. A. Pirwani, M. W. Vannier, J. S. Allen and H. Damasio, "Introduction to the non-rigid image registration evaluation project (NIREP)," in *Biomedical Image Registration, Proceedings, Vol. 4057*, (2006), pp. 128-135.
132. P. Hellier, C. Barillot, I. Corouge, B. Gibaud, G. Le Goualher, D. L. Collins, A. Evans, G. Malandain, N. Ayache, G. E. Christensen and H. J. Johnson, "Retrospective evaluation of intersubject brain registration," *Medical Imaging, IEEE Transactions on* **22**, 1120-1130 (2003).
133. T. Rohlfing, "Transformation model and constraints cause bias in statistics on deformation fields," *Med Image Comput Comput Assist Interv Int Conf Med Image Comput Comput Assist Interv* **9**, 207-214 (2006).
134. P. W. Roger, "Validation of registration accuracy," in *Handbook of medical imaging*, (Academic Press, Inc., 2000), pp. 491-497.
135. X. Geng, D. Kumar and G. E. Christensen, "Transitive inverse-consistent manifold registration," (Springer-Verlag, Glenwood Springs, CO, USA, 2005), pp. 468-479.
136. H. Jianchun, G. E. Christensen, J. T. Rubenstein and W. Ge, "Analysis of a new method for consistent large-deformation elastic image registration," *Vol. 4684*, (SPIE-Int. Soc. Opt. Eng, San Diego, CA, USA, 2002), pp. 945-954.
137. P. Hellier, C. Barillot, I. Corouge, B. Gibaud, G. Le Goualher, D. L. Collins, A. Evans, G. Malandain, N. Ayache, G. E. Christensen and H. J. Johnson, "Retrospective evaluation of intersubject brain registration," *IEEE Trans Med Imaging* **22**, 1120-1130 (2003).
138. C. Fox, *An introduction to the calculus of variations*. (Dover Publications, New York, 1987).

139. J. Weickert, K. J. Zuiderveld, B. M. ter Haar Romeny and W. J. Niessen, "Parallel implementations of AOS schemes: a fast way of nonlinear diffusion filtering," in *Image Processing, 1997. Proceedings., International Conference on, Vol. 3*, (1997), pp. 396-399 vol.393.
140. W. H. Press, "Numerical recipes : the art of scientific computing," xx, 818 p. (1986).
141. J. Weickert, "Recursive separable schemes for nonlinear diffusion filters," in *Scale-Space Theory in Computer Vision, Vol. 1252*, (Springer-Verlag Berlin, Berlin 33, 1997), pp. 260-271.
142. G. E. Christensen, M. I. Miller, M. W. Vannier and U. Grenander, "Individualizing neuroanatomical atlases using a massively parallel computer," *Computer* **29**, 32-& (1996).
143. M. Kass, A. Witkin and D. Terzopoulos, London, Engl, 1987 (unpublished).
144. W. Schroeder, K. Martin and B. Lorensen, *The visualization toolkit : an object-oriented approach to 3D graphics*, 4th ed. (Kitware, New York, 2006).
145. H. N. Christiansen and T. W. Sederberg, "CONVERSION OF COMPLEX CONTOUR LINE DEFINITIONS INTO POLYGONAL ELEMENT MOSAICS," **12**, 187-192 (1978).
146. M. Lutz, *Programming Python*, 3rd ed. (O'Reilly, Sebastopol, CA, 2006).
147. W. Lu, G. H. Olivera, Q. Chen, M. L. Chen and K. J. Ruchala, "Automatic re-contouring in 4D radiotherapy," *Phys Med Biol* **51**, 1077-1099 (2006).
148. K. S. Chao, S. Bhide, H. Chen, J. Asper, S. Bush, G. Franklin, V. Kavadi, V. Liengswangwong, W. Gordon, A. Raben, J. Strasser, C. Koprowski, S. Frank, G. Chronowski, A. Ahamad, R. Malyapa, L. Zhang and L. Dong, "Reduce in variation and improve efficiency of target volume delineation by a computer-assisted system using a deformable image registration approach," *Int J Radiat Oncol Biol Phys* **68**, 1512-1521 (2007).
149. J. Orban de Xivry, G. Janssens, G. Bosmans, M. De Craene, A. Dekker, J. Buijsen, A. van Baardwijk, D. De Ruyscher, B. Macq and P. Lambin, "Tumour delineation and cumulative dose computation in radiotherapy based on deformable registration of respiratory correlated CT images of lung cancer patients," *Radiother Oncol* (2007).
150. E. Rietzel, G. T. Chen, N. C. Choi and C. G. Willet, "Four-dimensional image-based treatment planning: Target volume segmentation and dose calculation in the presence of respiratory motion," *Int J Radiat Oncol Biol Phys* **61**, 1535-1550 (2005).
151. P. J. Keall, S. Joshi, S. S. Vedam, J. V. Siebers, V. R. Kini and R. Mohan, "Four-dimensional radiotherapy planning for DMLC-based respiratory motion tracking," *Med Phys* **32**, 942-951 (2005).

152. O. Dandekar, C. Castro-Pareja and R. Shekhar, "FPGA-based real-time 3D image preprocessing for image-guided medical interventions," *Journal of Real-Time Image Processing* **1**, 285-301 (2007).
153. O. Dandekar and R. Shekhar, "FPGA-accelerated deformable image registration for improved target-delineation during CT-guided interventions," *IEEE Transactions on Biomedical Circuits and Systems* **1**, 116-127 (2007).
154. O. Dandekar, V. Walimbe and R. Shekhar, "Hardware implementation of hierarchical volume subdivision-based elastic registration," (IEEE, New York, NY, USA, 2006), pp. 1425-1428.
155. C. R. Castro-Pareja, J. M. Jagadeesh and R. Shekhar, "FAIR: a hardware architecture for real-time 3-D image registration," *IEEE Trans Inf Technol Biomed* **7**, 426-434 (2003).
156. C. R. Castro-Pareja, J. M. Jagadeesh and R. Shekhar, "FPGA-based acceleration of mutual information calculation for real-time 3D image registration," *Vol. 5297*, (International Society for Optical Engineering, Bellingham, WA 98227-0010, United States, San Jose, CA, United States, 2004), pp. 212-219.
157. J. Jun, W. Luk and D. Rueckert, "FPGA-based computation of free-form deformations in medical image registration," (IEEE, Tokyo, Japan, 2003), pp. 234-241.
158. J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Maeurer and D. Shippy, "Introduction to the cell multiprocessor," *IBM Journal of Research and Development* **49**, 589-604 (2005).
159. M. Ohara, Y. Hangu, F. Savino, G. Iyengar, G. Leiguang, H. Inoue, H. Komatsu, V. Sheinin, S. Daijvaa and B. Erickson, "Real-time mutual-information-based linear registration on the Cell Broadband Engine Processor," (IEEE, Arlington, VA, USA, 2007), pp. 33-36.
160. G. L. Chen, G. Z. Sun, Y. Q. Zhang and Z. Y. Mo, "Study on parallel computing," *Journal of Computer Science and Technology* **21**, 665-673 (2006).
161. M. P. Wachowiak and T. M. Peters, "High-performance medical image registration using new optimization techniques," *IEEE Transactions on Information Technology in Biomedicine* **10**, 344-353 (2006).
162. T. Rohlfing and C. R. Maurer Jr, "Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees," *IEEE Transactions on Information Technology in Biomedicine* **7**, 16-25 (2003).
163. R. Chandra, *Parallel programming in OpenMP*. (Morgan Kaufmann Publishers, San Francisco, CA, 2001).
164. S. K. Warfield, F. A. Jolesz and R. Kikinis, "High performance computing approach to the registration of medical imaging data," *Parallel Computing* **24**, 1345-1368 (1998).

165. S. Ourselin, R. Stefanescu and X. Pennec, "Robust registration of multi-modal images: Towards real-time clinical applications," in *5th Int. Conf. MICCAI-Part II*, (2002), pp. 140-147.
166. F. Ino, Y. Tanaka, H. Kitaoka and K. Hagihara, "Performance study of nonrigid registration algorithm for investigating lung disease on clusters," *Vol. 2005*, (Institute of Electrical and Electronics Engineers Computer Society, Piscataway, NJ 08855-1331, United States, Dalian, China, 2005), pp. 820-825.
167. J. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Kruger, A. Lefohn and T. Purcell, "A Survey of General-Purpose Computation on Graphics Hardware," in *Eurographics 2005, State of the Art Reports*, (2005), pp. 21-51.
168. M. Woo and OpenGL Architecture Review Board., *OpenGL programming guide : the official guide to learning OpenGL, version 1.2*, 3rd ed. (Addison-Wesley, Reading, MA, 1999).
169. J. Sanchez and M. P. Canton, *DirectX 3D graphics programming bible*. (IDG Books Worldwide, Foster City, CA, 2000).
170. R. Fernando and M. J. Kilgard, *The Cg tutorial : the definitive guide to programmable real-time graphics*. (Addison-Wesley, Boston, Mass. ; London, 2003).
171. P. Walsh, *Advanced visual effects with direct 3d*. (Thomson Course Technology, Boston, MA, 2006).
172. NVIDIA Corp., "<http://developer.nvidia.com/object/cuda.html>," (2007).
173. AMD Inc., "[http://ati.amd.com/companyinfo/researcher/documents/ATI\\_CTM\\_Guide.pdf](http://ati.amd.com/companyinfo/researcher/documents/ATI_CTM_Guide.pdf)," (2006).
174. I. Buck, T. Foley, D. Horn, J. Sugerman, K. Fatahalian, M. Houston and P. Hanrahan, "Brook for GPUs: Stream computing on graphics hardware," *Vol. 23*, (Association for Computing Machinery, Los Angeles, CA, United states, 2004), pp. 777-786.
175. Intel Corp., "<http://www.intel.com/performance/server/xeon/hpcapp.htm>."
176. NVIDIA Corp., "NVIDIA GeForce 8800 GPU Architecture Overview," Technical Brief (2006).
177. J. Spoerk, H. Bergmann, F. Wanschitz, S. Dong and W. Birkfellner, "Fast DRR splat rendering using common consumer graphics hardware," *Medical Physics* **34**, 4302-4308 (2007).
178. A. Khamene, P. Bloch, W. Wein, M. Svatos and F. Sauer, "Automatic registration of portal images and volumetric CT for patient positioning in radiation therapy," *Medical Image Analysis* **10**, 96-112 (2006).

179. H. Hong, K. Kim and S. Park, "Fast 2D-3D point-based registration using GPU-based preprocessing for image-guided surgery," in *Progress in Pattern Recognition, Image Analysis and Applications, Proceedings, Vol. 4225*, (Springer-Verlag Berlin, Berlin, 2006), pp. 218-226.
180. C. Vetter, C. Guetter, C. Xu and R. Westermann, "Non-rigid multi-modal registration on the GPU," *Vol. 6512*, (SPIE, Bellingham WA, WA 98227-0010, United States, San Diego, CA, United States, 2007), pp. 651228.
181. G. C. Sharp, N. Kandasamy, H. Singh and M. Folkert, "GPU-based streaming architectures for fast cone-beam CT image reconstruction and demons deformable registration," *Phys Med Biol* **52**, 5771-5783 (2007).
182. S. Grzegorz, B. Michael, H. Peter, N. Christopher, G and G. nther, "Non-rigid Registration with Use of Hardware-Based 3D Bezier Functions," in *Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention-Part II*, (Springer-Verlag, 2002).
183. P. Hastreiter, C. Rezk-Salama, G. Soza, M. Bauer, G. Greiner, R. Fahlbusch, O. Ganslandt and C. Nimsky, "Strategies for brain shift evaluation," *Med Image Anal* **8**, 447-464 (2004).
184. C. Rezk-Salama, M. Scheuering, G. Soza and G. Greiner, "Fast volumetric deformation on general purpose hardware," in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, (ACM, Los Angeles, California, United States, 2001).
185. D. Levin, D. Dey and P. J. Slomka, "Acceleration of 3D, nonlinear warping using standard video graphics hardware: implementation and initial validation," *Computerized Medical Imaging and Graphics* **28**, 471-483 (2004).
186. R. Strzodka, M. Droske and M. Rumpf, "Image registration by a regularized gradient flow. A streaming implementation in DX9 graphics hardware," *Computing (Vienna/New York)* **73**, 373-389 (2004).
187. A. Köhn, J. Drexler, F. Ritter, M. König and H. Peitgen, "GPU Accelerated Image Registration in Two and Three Dimensions," in *Bildverarbeitung für die Medizin 2006*, (2006), pp. 261-265.
188. J. Kim, S. Li, Y. Zhao and B. Movsas, "Real-time intensity-based deformable fusion on PC graphics hardware," In *XVth International Conference on the Use of Computers in Radiation Therapy* (2007).
189. N. Courty and P. Hellier, "Accelerating 3D Non-Rigid Registration using Graphics Hardware," *International Journal of Image and Graphics* **8**, 1-18 (2008).

190. J. D. Owens, D. Luebke, N. Govindaraja, M. J. Harris, J. Kruger, A. E. Lefohn and T. J. Purcell, "A survey of general-purpose computation on graphics hardware," in *Eurographics, State of the Art Reports*, (2005), pp. 21-51.

## BIOGRAPHICAL SKETCH

Junyi Xia was born in 1974 in Dinghai, ZheJiang, People's Republic of China. Junyi graduated from ZheJiang University with a Bachelor of Science degree in chemical machinery in 1996. After two year's graduate study, Junyi was graduated with a Master of Science degree in chemical machinery. Upon graduation, he worked as a software engineer in Supcon Group Ltd. China. In 2000, Junyi was enrolled in the graduation program in the Department of Biomedical Engineering at the University of Memphis, where he served as the chief mechanical designer under the supervision of Dr.Sanjiv Samant. In 2003, Junyi was graduated with a Master of Science degree in biomedical engineering.

In 2005, Junyi joined the graduate program in the Department of Nuclear and Radiological Engineering at the University of Florida to pursue his Ph.D with Dr. Sanjiv Samant in the field of image guided radiation therapy. During his Ph.D studies, Junyi presented invited speeches in numerous national and international conferences and served as reviewers for scientific journals and international imaging conferences. Junyi married Yuming Zhao, on June 9<sup>th</sup> 2006.