2013

Adaptive Stochastic Conjugate Gradient optimization for temporal medical image registration

Huanhuan Xu
Louisiana State University and Agricultural and Mechanical College

ADAPTIVE STOCHASTIC CONJUGATE GRADIENT OPTIMIZATION FOR TEMPORAL MEDICAL IMAGE REGISTRATION

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science

in

The Department of Mathematics

by
Huanhuan Xu
B.S., Central China Normal University, 2006
M.S., University of Science and Technology of China, 2009
Ph.D., Louisiana State University, 2013
December 2013

Acknowledgments

I would like to express my deepest gratitude to my advisors Professor Hongchao Zhang and Professor Xin Li for their guidance and support during this research project. I also want to thank Professor Blaise Bourdin for being my committee. I would like to thank Professor Shawn Walker for many valuable discussions and University of Texas Southwestern Medical Center for providing the real-patient data.

This thesis is dedicated to my family for their love and encouragement.

Table of Contents

Acknowledgments ii

List of Tables iv

List of Figures v

Abstract vi

Chapter 1: Introduction 1
- 1.1 Spatio-temporal Image Registration Problem 1
- 1.2 Solving the Optimization 6
- 1.3 Organization 7

Chapter 2: General Optimization Methods 8
- 2.1 Steepest Gradient Descent 9
- 2.2 Quasi-Newton 10
- 2.3 Nonlinear Conjugate Gradient 12

Chapter 3: Adaptive Stochastic Conjugate Gradient 14
- 3.1 Adaptive Stochastic Gradient Descent 14
- 3.2 Adaptive Stochastic Conjugate Gradient 17

Chapter 4: Experimental Results 20
- 4.1 Experiment Setup 20
- 4.2 The Choice of the Optimum Scheme for ASCG 21
- 4.3 ASCG with Different Image Resolution 22
- 4.4 ASCG with Image Noise 24
- 4.5 Optimization Comparison 25
- 4.6 Application: Motion Modeling of Clinical Lung Tumor Scans 27

Chapter 5: Conclusion 28

References 29

Vita 31

List of Tables

4.1 The landmark prediction error $D_{1,6}$ and its standard deviation $\sigma_{1,6}$ (in mm) with different search direction and step size. This is the registration between I_1 and I_6 on the POPI data. 22

4.2 The landmark predication error D_i and its standard deviation σ_i (in mm) of i^{th} time frame on the POPI-data [1] with different optimization methods. $\bar{D}$ is the average MTRE. 26

4.3 The landmark prediction error D and its standard deviation σ (in mm) on the DIR-LAB data set with different optimization methods. 26

List of Figures

1.1 The basic registration components. 1

1.2 Mapping model. 2

3.1 Examples of the sigmoid function with different ω. $f_{MAX} = 1$ and $f_{MIN} = -0.5$. 16

4.1 The illustration of MTRE between two points in the images I_r and I_t . . . 21

4.2 The registration accuracy w.r.t. the coefficient α under different image resolution. 23

4.3 Time comparison with different image resolution. 24

4.4 The 2D cross section of the 4D images. 25

4.5 Registration accuracy with respect to the image noise under different optimization schemes. 26

4.6 Lung/Tumor Tracking via a Deforming Surface Geometry. (a, b) show the alignment of iso-contours and the scanned images. (d) shows the color-coded displacement field of $T^{1,6}(I_1)$ from I_1 in (c); (e) visualizes the Hausdorff distance from the deformable model to the scan. 27

Abstract

We propose an Adaptive Stochastic Conjugate Gradient (ASCG) optimization algorithm for temporal medical image registration. This method combines the advantages of Conjugate Gradient (CG) method and Adaptive Stochastic Gradient Descent (ASGD) method. The main idea is that the search direction of ASGD is replaced by stochastic approximations of the conjugate gradient of the cost function. In addition, the step size of ASCG is based on the approximation of the Lipschitz constant of the stochastic gradient function. Thus, this algorithm could maintain the good properties of the conjugate gradient method, meanwhile it uses less gradient computation time per iteration and adjusts the step size adaptively as the ASGD method. As a result, this algorithm takes less CPU time than the previous ASGD method.

We demonstrate the efficiency of our algorithm on the public available 4D Lung CT data and our clinical Lung/Tumor CT data using the general 4D image registration model. We compare the ASCG with several existing iterative optimization strategies: steepest gradient descent method, conjugate gradient method, Quasi-Newton method (LBFGS) and adaptive stochastic gradient descent method. Our preliminary results indicate that our ASCG algorithm achieves 22% higher accuracy on the POPI dataset and it also performs better than existing methods on other datasets(DIR-Lab dataset and our clinical dataset). Furthermore, we demonstrate that compared with other methods, our ASCG algorithm is more robust to image noises.

1 Introduction

This chapter introduces a symmetric 4D registration model we are trying to solve in this work. We give the specific definition and optimization strategy of this model, which is treated as a nonlinear optimization problem. At the end of this chapter, we provide an outline of this thesis.

1.1 Spatio-temporal Image Registration Problem

Image registration is important in medical image analysis. For example, in lung cancer radiotherapy, it can establish the temporal correspondences among the scanned 4D (temporally sequential volume) CT images, for building the motion estimation model to describe the movement and deformation of organs during respiratory cycles. Figure 1.1 shows the general components of an image registration algorithm in a block scheme. Usually two images are involved in the registration process. One image, the moving image I_M, is deformed to fit the other image, the fixed image I_F. The fixed and moving image are of dimension D

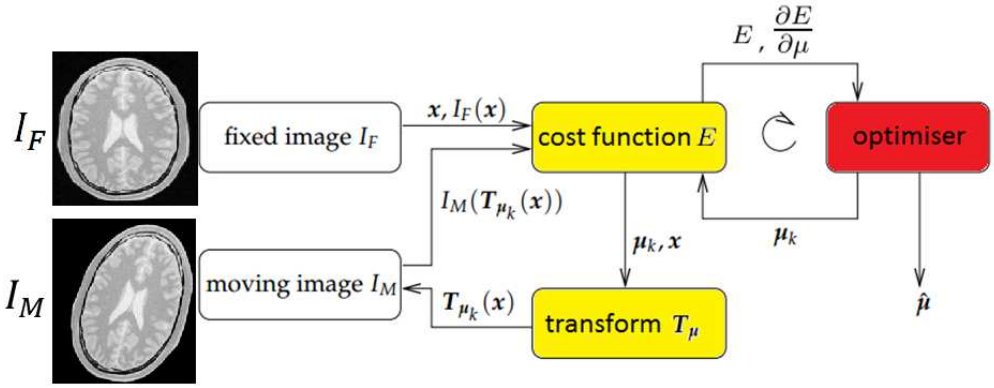


FIGURE 1.1. The basic registration components.

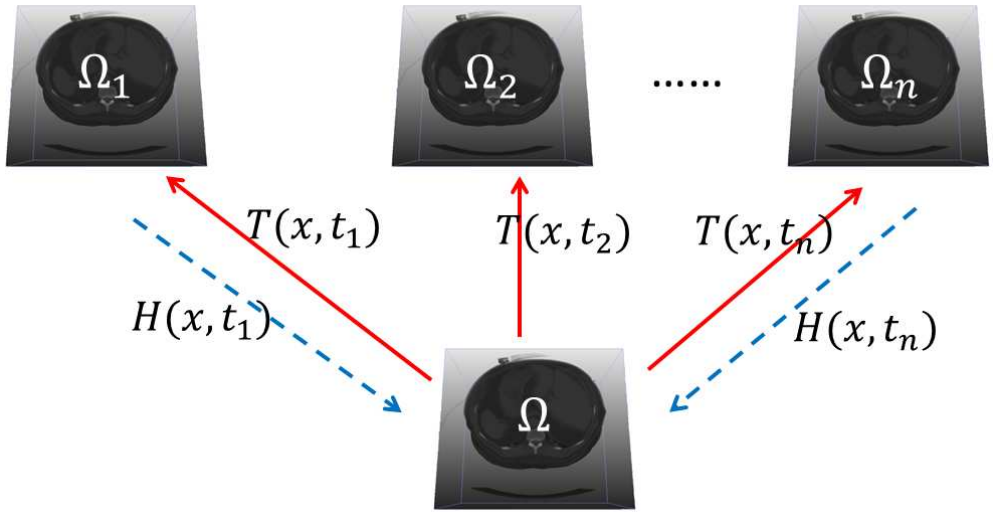


FIGURE 1.2. Mapping model.

and are defined on their own spatial domain: $\Omega_F \subset \mathcal{R}^D$ and $\Omega_M \subset \mathcal{R}^D$ respectively, where Ω_F, Ω_M are the nonempty, bounded, open sets in $\mathcal{R}^D$. Registration is the problem of finding a transformation T that makes $I_M(T)$ spatially aligned to I_F. The transformation is defined as a mapping from the fixed image to the moving images, i.e. $T : \Omega_F \to \Omega_M$. The quality of alignment is defined by a cost function E. Commonly, the registration problem is formulated as an optimization problem in which the cost function is minimized with respect to T. Thus the optimizer is very important in this framework, which adjusts the transform parameters to minimize the cost function. The good optimizer should be reliable and can find the best solution quickly.

In this work, we are going to solve a complex 4D image registration problem. Given a sequence of volume images, $I_1, I_2, \ldots, I_\Gamma$, where each image $I_i(\mathbf{x}) : \Omega_i \to \mathcal{R}, \mathbf{x} \in \Omega_i \subset \mathcal{R}^3$ is a 3D intensity function[1], we want to compute a temporally deforming 3D model $T(\mathbf{x}, t) : \Omega \times \mathcal{R} \to \mathcal{R}^3, \Omega \subset \mathcal{R}^3$ that correlates all the input images, as illustrated in Figure 1.2. A point $\mathbf{x} \in \Omega_i$ in I_i is correlated with a point $\mathbf{x}'$ in I_j by $\mathbf{x}' = T(T^{-1}(\mathbf{x}, t_i), t_j)$. Then, a continuous

[1]For sequential CT scans, their parametric domains Ω_i simply overlay in $\mathcal{R}^3$

4D deforming image $I(\mathbf{x}, t)$ can be constructed using the intensity function defined in the first image I_1, namely, $I(\mathbf{x}, t) = I_1(T(T^{-1}(x, t), t_1))$.

To obtain this deforming parametric geometry and the deforming image, we need to explicitly compute two 4D functions: (1) a forward 4D parameterization T, spatially defined on a common parametric domain, $T : \Omega \times \mathcal{R} \rightarrow \mathcal{R}^3$, and (2) its inverse mapping $H = T^{-1} : \mathcal{R}^3 \times \mathcal{R} \rightarrow \Omega$ which maps coordinate space of the deforming images $\Omega_i (\subset \mathcal{R}^3) \times \mathcal{R}$ to the common domain. To model the nonrigid freeform deformations of human organs during respiratory cycles, we use 4D B-spline functions to approximate these two transformations T and H, through which both the spatial and temporal smoothness can be formulated easily. The B-spline approximation for T can be formulated as:

$$T(\mathbf{y}) = \mathbf{x} + \sum_{\mathbf{y}_k \in N_{\mathbf{y}}} p_k \beta^r(\mathbf{y} - \mathbf{y}_k), \tag{1.1}$$

where $\mathbf{y} = (\mathbf{x}, t)$, $\mathbf{y}_k$ is a knot on the parametric domain $\Omega \times \mathcal{R}$; $\beta^r(\cdot)$ is the r-th order multidimensional B-spline polynomial (here we take $r = 3$); $\mathbf{p}_k \in \mathcal{R}^3$ are B-spline control points to be solved, and $N_{\mathbf{y}}$ denotes $\mathbf{y}$'s neighboring local support regions where the basis functions are nonzero. The knots $\mathbf{y}_k$ are defined on a 4D regular grid, uniformly overlaid the 4D image.

Because the inverse of B-spline transformation cannot be derived in close-form, we explicitly approximate this inverse mapping using another B-spline transformation H using a same formulation to eq (1.1). Then with T and H, a transformation T^{ij} from any frames i to j can be composed as

$$T^{ij}(\mathbf{x}) = T(H(\mathbf{x}, t_i), t_j), \mathbf{x} \in \Omega_i. \tag{1.2}$$

The entire 4D registration problem is formulated as an optimization on T and H that minimizes an objective function:

$$E = E_I + \alpha E_F + \lambda E_S + \rho E_C, \tag{1.3}$$

where E_I measures the intensity matching error, E_F measures the feature alignment, E_S measures the spatial and temporal smoothness of the deformation, E_C measures the inverse consistency, and α, λ, ρ are weighting factors.

Intensity Matching Error. With the assumption that the corresponded points have the same intensity, the registration should minimize the intensity differences of corresponded points. We can derive the intensity difference between corresponded points in any pair of images I_i and I_j taken in time t_i and t_j. For any point $\mathbf{x} \in \Omega_i$ in time t_i, its corresponding location in time t_j can be composed by H and T. The accumulated difference between $I_i(\mathbf{x})$ and the intensity of its corresponding coordinate in t_j can be formulated as:

$$\tilde{E}_I = \frac{1}{|S||\Gamma|^2} \sum_{t_i \in \Gamma} \sum_{t_j \in \Gamma} \sum_{\mathbf{x} \in S_i} (I_j(T(H(\mathbf{x}, t_i), t_j)) - I_i(\mathbf{x}))^2, \tag{1.4}$$

where S_i is the sets of spatial voxel coordinates in each Ω_i and for $\forall i, |S| = |S_i|$. Simultaneously solving both T and H is expensive. We first solve a forward parameterization T, then iteratively, fix the parameterization in one direction and optimize the other (see Section 1.2 for the complete algorithm).

To solve the initial forward parameterization T without knowing H, we formulate the reduction of intensity error by minimizing the intensity variance:

$$T_I = \frac{1}{|S||\Gamma|} \sum_{\mathbf{x} \in S} \sum_{t \in \Gamma} (I_t(T(\mathbf{x}, t)) - \bar{I}(\mathbf{x}))^2, \tag{1.5}$$

where $\bar{I}(\mathbf{x})$ is the average intensity value follows the forward parameterization: $\bar{I}(\mathbf{x}) = \frac{1}{|\Gamma|} \sum_{t \in \Gamma} I_t(T(\mathbf{x}, t))$. $S \subset \Omega$ are the spatial voxel coordinates (e.g. coordinates of all the pixels) and $\Gamma \subset \mathcal{R}$ contains the temporal coordinates indexing temporal sample images. After obtaining the initial T, we iteratively optimize H and T by minimizing:

$$E_I = T_I + \tilde{E}_I. \tag{1.6}$$

Feature Alignment Error. The intensity term has many local minima. Geometric features can help effectively avoid many undesirable solutions. We extract feature points using a slightly modified 3D SIFT algorithm [2], then compute a set of consistently corresponded feature points $\{p_{ij}\}$ across the entire sequence of images, where p_{ij} indicates the i-th feature point on time t_j, where $i = 1, \ldots, N, j = 1, \ldots, |\Gamma|$.

Each consistently corresponded feature point has a parametric coordinate $m_i, i = 1, \ldots, N$ in Ω, which is mapped to the feature p_{it} in image I_t at time t. The feature correspondence in the forward parameterization should penalize the deviation of $T(m_i, t)$ from p_{it}:

$$T_F = \frac{1}{N|\Gamma|} \sum_{t\in\Gamma} \sum_{i=1}^{N} ||p_{it} - T(m_i, t)||^2, \tag{1.7}$$

For the inverse map H, the variance of $H(p_{ij}, j)$ should be minimized:

$$H_F = \frac{1}{N|\Gamma|} \sum_{i=1}^{N} \sum_{t\in\Gamma} ||H(p_{it}, t) - \bar{H}(p_{i*})||, \tag{1.8}$$

where $\bar{H}(p_{i*}) = \frac{1}{|\Gamma|} \sum_{t\in\Gamma} H(p_{i,t}, t)$ is the average coordinates of the i-th feature p_{i*}. Finally, the entire feature alignment error is:

$$E_F = T_F + H_F. \tag{1.9}$$

Deformation and Motion Smoothness. The transformation (hence both parameterizations T and H) should be spatially and temporally smooth. The 2nd-order derivatives of the B-spline transformation functions can be derived as the smoothness energy to minimize:

$$\begin{aligned} E_S &= T_S + H_S; \\ T_S &= \tfrac{1}{|S||\Gamma|} \textstyle\sum_{\mathbf{x}\in S} \sum_{t\in\Gamma} ||\frac{\partial^2 T}{\partial \mathbf{y} \partial \mathbf{y}^T}||_F^2, \mathbf{y} = (\mathbf{x}, t) \\ H_S &= \tfrac{1}{|S||\Gamma|} \textstyle\sum_{\mathbf{x}\in S_i} \sum_{t\in\Gamma} ||\frac{\partial^2 H}{\partial \mathbf{y} \partial \mathbf{y}^T}||_F^2, \mathbf{y} = (\mathbf{x}, t); \end{aligned} \tag{1.10}$$

where

$||\frac{\partial^2 T}{\partial \mathbf{y} \partial \mathbf{y}^T}||_F^2 = \sum_{j=1}^{3}((\frac{\partial^2 T_j}{\partial x_1^2})^2+(\frac{\partial^2 T_j}{\partial x_2^2})^2+(\frac{\partial^2 T_j}{\partial x_3^2})^2+(\frac{\partial^2 T_j}{\partial t^2})^2+2(\frac{\partial^2 T_j}{\partial x_1 \partial x_2})^2+2(\frac{\partial^2 T_j}{\partial x_1 \partial x_3})^2+2(\frac{\partial^2 T_j}{\partial x_2 \partial x_3})^2+2(\frac{\partial^2 T_j}{\partial x_1 \partial t})^2+2(\frac{\partial^2 T_j}{\partial x_2 \partial t})^2+2(\frac{\partial^2 T_j}{\partial x_3 \partial t}))^2, \mathbf{y} = (x_1, x_2, x_3, t)$.

Inverse Consistency means the matching between two frames I_i and I_j should be symmetric and bijective. Namely, the matching from I_i to I_j (composed by T and H) is one-to-one and also consistent with that from I_j to I_i. In 3D pairwise registration, inverse consistency also means the selection of reference doesn't affect the matching result [3]. This consistency can be achieved by making the composition of T and H to be as near-identity as possible:

$$E_C = \frac{1}{|S||\Gamma|}\sum_{\mathbf{x}\in S_i}\sum_{t\in\Gamma}||T(H(\mathbf{x},t),t)-\mathbf{x}||^2 + \frac{1}{|S||\Gamma|}\sum_{\mathbf{x}\in S}\sum_{t\in\Gamma}||H(T(\mathbf{x},t),t)-\mathbf{x}||^2, \quad (1.11)$$

1.2 Solving the Optimization

Simultaneously solving T and H reduces to a very expensive optimization problem. We develop an iterative algorithm to seek for the optimal solution. During each iteration, T (or H) is solved using a gradient-based optimization algorithm ASCG (Adaptive Stochastic Conjugate Gradient) we proposed in this work, which is the combination of the conjugate gradient [4] and adaptive stochastic gradient method [5]. With the B-spline representation we derive the derivatives of E_F, E_S, E_C explicitly, and we use the finite difference approximation to get the derivatives of E_I.

We first solve a forward parameterization T by minimizing $T_I + \alpha T_F + \rho T_S$ from equations (1.5,1.7,1.10), then with T fixed, we solve its inverse parameterization H by minimizing the entire objective function E in equation (1.3). Then iteratively, we fix one parameterization and revise its inverse parameterization, until the energy reduction is smaller than a threshold. This optimization algorithm is formulated as follows.

Algorithm 1:

1) Compute an initial forward parameterization T by minimizing $T_I + \alpha T_F + \rho T_S$;

2) Fix T, and solve H by minimizing E;

3) Fix H, and solve T by minimizing E;

4) If E converges, STOP; otherwise GOTO 2).

1.3 Organization

In chapter 2 we provide a brief introduction of several existing gradient descent methods which will be used to compare with our proposed method ASCG in the experiments part. In chapter 3 we give the detail formulation of ASCG and its convergence analysis. The experiments and results are described in chapter 4. The optimization methods are tested on public available Lung CT data and our clinical Lung/Tumor CT data. Conclusions are given in chapter 5.

2 General Optimization Methods

We let μ be a vector concatenating all unknown control points which becomes a large dimension unknown parameters vector. In order to get one 4D parameterization during each step in Algorithm 1, we are trying to solve following optimization problem.

$$\hat{\mu} = \arg\min_{\mu} E(\mu), \tag{2.1}$$

The solution $\hat{\mu}$ is the solution that minimizes the objective function $E(\mu)$. The objective function E may have multiple local minima. Which local minimum is selected as the solution $\hat{\mu}$ depends on the optimization algorithm and on the initial guess. To determine the optimal set of parameters $\hat{\mu}$ an iterative optimization strategy is employed:

$$\mu_{k+1} = \mu_k + \lambda_k d_k, \;\; k = 0, 1, \ldots, K, \tag{2.2}$$

where d_k is the search direction at iteration k, and λ_k is a scalar gain factor controlling the step size along the search direction. The search directions and gain factors are chosen such that the sequence $\{\mu_k\}$ converges to a local minimum of the objective function E. Many optimization methods can be found in the literature [6], differing in the way λ_k and d_k are computed.

In this work, several optimization methods are compared with respect to speed, accuracy, precision, and robustness. The following methods are included in the study: steepest gradient descent [6], conjugate gradient method [7, 8], Quasi-Newton method (BFGS [9], LBFGS [10]), adaptive stochastic gradient descent [5]. The first three are deterministic gradient-based

algorithms. They have in common that the expression for the search direction d_k is based on $\frac{\partial E}{\partial \mu}$, the derivative of the cost function with respect to the parameters, and they assume that $\frac{\partial E}{\partial \mu}$ can be computed exactly. The last method is stochastic gradient-based algorithms. They also derive search directions from $\frac{\partial E}{\partial \mu}$, but only need stochastic approximations of the derivative, potentially faster to compute than the exact derivative.

Many strategies exist for determining the step size λ_k. It can, for example, simply be set to a constant, or defined by a decaying function of k. Another possibility is the use of a line search, which, in each iteration, tries to minimize the cost function E along the search direction d_k:

$$\lambda_k = \arg\min_{\lambda} E(\mu_k + \lambda d_k). \tag{2.3}$$

The disadvantages of such an exact line search are that many additional evaluations of the cost function and/or its derivative are required. Therefore, an inexact line search is more often used. Instead of solving eq.(2.3) exactly, an inexact line search finds a gain factor λ_k that gives a sufficient reduction of E.

In all but one of the investigated optimization methods, the expression for d_k is based on the derivative of the objective function, $\frac{\partial E}{\partial \mu}$, henceforth referred to as g. And an analytic expression for the derivative of the cost function E is available. Some optimization methods require exact evaluation of this expression. Other methods are satisfied with an approximation [11].

2.1 Steepest Gradient Descent

The gradient descent method [6] takes steps in the direction of the negative gradient of the cost function:

$$\mu_{k+1} = \mu_k - \lambda_k g(\mu_k), \tag{2.4}$$

where $g(\mu_k)$ is the derivative of the objective function evaluated at the current position μ_k.

In this work, the gain factor λ_k is determined by an inexact line search routine described by More and Thuente [12]. It determines λ_k such that the so-called strong Wolfe conditions are satisfied:

$$E(\mu_{k+1}) \leq E(\mu_k) + c_1 \lambda_k d_k^T g(\mu_k) \tag{2.5}$$

$$|d_k^T g(\mu_{k+1})| \leq c_2 |d_k^T g(\mu_k)| \tag{2.6}$$

with user-defined scalars c_1 and c_2 satisfying $0 < c_1 < c_2 < 1$. Recall that d_k represents the search direction of the optimization algorithm. The first Wolfe condition 2.5 demands a sufficient decrease of the cost function value. The second Wolfe condition 2.6 enforecs reasonable progress towards a stationary point of the cost function, where the derivative vanishes.

In order to give an indication of the rate of convergence of gradient descent methods, it is possible to derive theoretical bounds on the distance to the solution at iteration k, $||\mu_k - \hat{\mu}||$. Provided that the sequence $\{\mu_k\}$ converges to a local nonsingular minimum $\hat{\mu}$ of E, it can be proven [13] that there exist a $K \geq 0$ and $0 < \rho < 1$ such that the following expression holds:

$$\frac{||\mu_{k+1} - \hat{\mu}||}{||\mu_k - \hat{\mu}||} \leq \rho, \text{ for all } k \geq K. \tag{2.7}$$

This means that the method has a linear rate of convergence. This method is sensitive to poor scaling and converges slow.

2.2 Quasi-Newton

Before Quasi-Newton methods [9], the well-known Newton method is given by:

$$\mu_{k+1} = \mu_k - [H(\mu_k)]^{-1} g(\mu_k), \tag{2.8}$$

where $H(\mu_k)$ is the Hessian matrix of the objective function, evaluated at μ_k. The use of such second-order information gives the algorithm better theoretical convergence properties than the gradient descent. The computation of the Hessian matrix and its inverse is computationally expensive, especially in the 4D image registration problem. Quasi-Newton methods tackle this problem by using an approximation to the inverse of the Hessian: $L_k \approx [H(\mu_k)]^{-1}$. The approximation is updated in every iteration k. Second-order derivatives of the objective function are not needed for this update; only the already computed first-order derivatives are used. Direct approximation of the inverse of the Hessian avoids the need for a matrix inversion. The gain factor (step-size) λ_k is determined by the inexact line search algorithm. This results in the following Quasi-Newton algorithm:

$$\mu_{k+1} = \mu_k - \lambda_k L_k g(\mu_k). \tag{2.9}$$

Given certain conditions, many Quasi-Newton methods can be shown to be superlinearly convergent [6]

$$\lim_{k \to \infty} \frac{||\mu_{k+1} - \hat{\mu}||}{||\mu_k - \hat{\mu}||} \to 0. \tag{2.10}$$

Many ways to construct the series L_k are proposed in the literature [6], most notably Symmetric-Rank-1 (SR1), Davidon-Fletcher-Powell(DFP), and Broyden-Fletcher-Goldfarb-Shanno (BFGS). Numerical experiments indicate that BFGS is very efficient in many applications. It uses the following update rule for L_k:

$$L_{k+1} = (I - \frac{s_k y_k^T}{s_k^T y_k}) L_k (I - \frac{y_k s_k^T}{s_k^T y_k}) + \frac{s_k s_k^T}{s_k^T y_k} \tag{2.11}$$

where I is the identity matrix, $s_k = \mu_{k+1} - \mu_k$, and $y_k = g_{k+1} - g_k$. The step-size λ_k is determined by the strong Wolfe conditions. To realize superlinear convergence it is important to always try a gain factor $\lambda_k = 1$ first [6]. If this step size does not satisfy the strong Wolfe conditions, the iterative inexact line search procedure is started to find a suitable gain. If no gain factor satisfying the strong Wolfe conditions can be found, the optimization is stopped. The limited memory version of BFGS method (LBFGS) eliminates the need for storing the matrix L_k in memory.

The main weakness of the LBFGS method are that it often converges slowly, which usually leads to a relatively large number of function evaluations, and that it is inefficient on highly ill-conditioned problems — specifically, on problems where the Hessian matrix contains a wide distribution of eigenvalues. Also this method requires high accurate gradient computation.

2.3 Nonlinear Conjugate Gradient

Before nonlinear conjugate gradient method, there is the linear conjugate gradient method which was designed for solving a system of linear equations. That is equivalent to the minimization of a quadratic objective function. The nonlinear conjugate gradient method is an extension suitable for minimizing general nonlinear functions [6, 4]. The search direction d_k of nonlinear conjugate gradient algorithm is defined as a linear combination of the gradient $g(\mu_k)$ and the previous search direction d_{k-1}:

$$d_k = -g_k + \beta d_{k-1}, \text{for } k = 1, ... \tag{2.12}$$

where $d_0 = -g_0$.

Several expressions for the scalar β have been proposed in the literature, including

$$\beta^{FR} = \frac{||g_k||^2}{||g_{k-1}||^2}, \tag{2.13}$$

$$\beta^{PRP} = \frac{g_k^T(g_k - g_{k-1})}{||g_{k-1}||^2}. \tag{2.14}$$

The choice of β has a large influence on the global convergence properties. For an extensive review on this topic, we refer to [8]. Numerical experience indicates that β^{PRP} tends to be more robust and efficient of the two [6]. Thus in our study, we use β^{PRP}.

Depending on the line search technique used, various theoretical bounds on the rate of convergence have been derived in the literature. In [14], it shows that with an inexact line search routine, a n-step linear rate of convergence can be achieved. However, the weakness of this method is that it requires more iterations in line search algorithm.

3 Adaptive Stochastic Conjugate Gradient

In this chapter, we formulate our proposed optimization method named Adaptive Stochastic Conjugate Gradient (ASCG) method. Before this, we present the current Adaptive Stochastic Gradient Descent (ASGD) method since ASCG is inspired by the ASGD.

3.1 Adaptive Stochastic Gradient Descent

The stochastic gradient descent method [15] follows the same scheme as the deterministic gradient descent method introduced in previous chapter, but with different schemes to decide the step-size λ_k and search direction d_k. The method uses the following iterative scheme:

$$\mu_{k+1} = \mu_k - \gamma(t_k)\tilde{g}_k, \;\; k = 0, 1, ..., K, \tag{3.1}$$

$$\tilde{g}_k = g(\mu_k) + \varepsilon_k, \tag{3.2}$$

where $\tilde{g}_k$ denotes an approximation of the true derivative g_k at μ_k, and ε_k is the approximation error. If $\varepsilon_k = 0$, Eq 3.1 equals the deterministic gradient descent procedure, described in the previous chapter. The approximation of $g(\mu_k)$ is realized by computing g using not all voxels, but only a small subset of voxels, randomly selected in every iteration. Convergence to the solution $\hat{\mu}$ can only be guaranteed [15] if the bias of the approximation error goes to zero

$$E(\tilde{g_k}) = g(\mu_k), \tag{3.3}$$

where $E(.)$ denotes expectation.

A stochastic gradient descent method is often applied when computation of the exact gradient is very costly or the exact gradient is not available. Using an approximation of the exact gradient could decrease the computation time per iteration, but may have negative effects on the speed of convergence.

The step size $\gamma(t_k)$ is determined by a predefined decaying function of the iteration number k. For example:

$$\gamma(t_k) = \frac{a}{(t_k + A)^\alpha} \tag{3.4}$$

$$t_{k+1} = [t_k + f(-\tilde{g}_k^T \tilde{g}_{k-1})]^+, \tag{3.5}$$

where $[x]^+$ means $max(x, 0)$, f denotes a sigmoid function with $f(0) = 0$, and

$$f(x) = f_{MIN} + \frac{f_{MAX} - f_{MIN}}{1 - (f_{MAX}/f_{MIN})e^{-x/\omega}}, \tag{3.6}$$

with $f_{MAX} > 0, f_{MIN} < 0$, and $\omega > 0$ which is user-specified. Examples of f are shown in Fig. 3.1. If $\omega \to 0$, the sigmoid function f approaches a step function.

$a > 0, A \geq 1$, and $0 < \alpha \leq 1$ are user-specified constants. A choice of $\alpha = 1$ gives a theoretically optimum rate of convergence when $k \to \infty$ [15]. In practice, the algorithm is stopped after a specified maximum number of iterations. The factor a is especially difficult to choose, since it has no unit, and heavily depends on the choice of the objective function. For example, when we multiple the objective function by an arbitrary constant c, the value of a would need to be divided by c in order to get the same sequence $\{\mu_k\}$. When a is set too small, the method suffers from slow convergence. When a is set too large, the process may become unstable.

Here the γ function is evaluated at the time t_k. The time is adapted depending on the inner product of the gradient $\tilde{g}_k$ and the previous gradient $\tilde{g}_{k-1}$. If the gradients in two consecutive steps point in the same direction, the inner product is positive, and therefore the time is

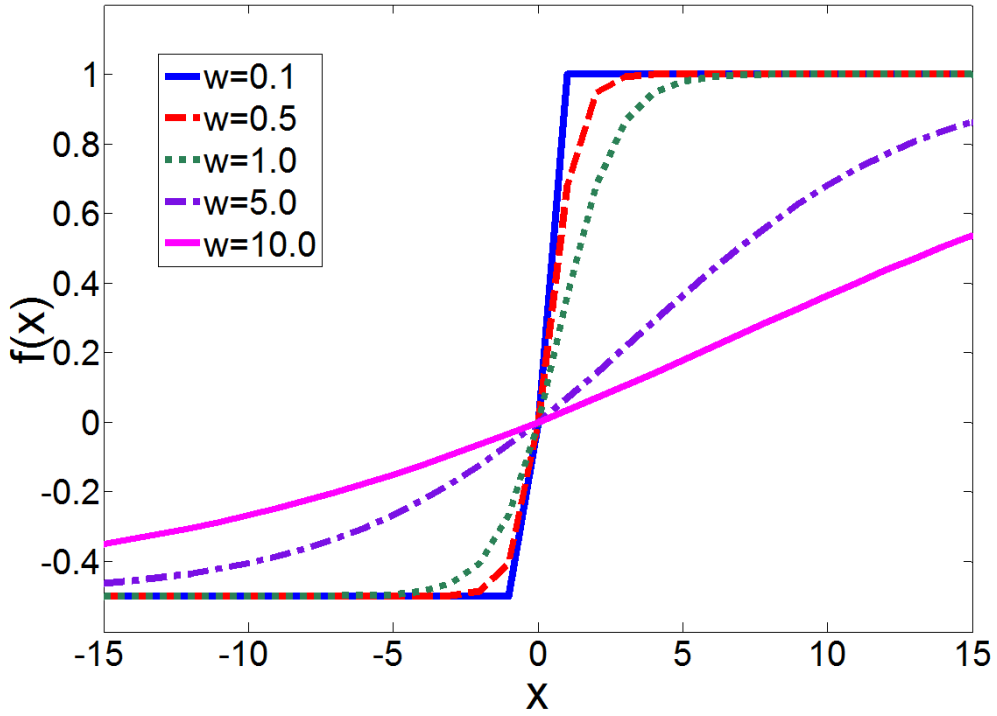


FIGURE 3.1. Examples of the sigmoid function with different ω. $f_{MAX} = 1$ and $f_{MIN} = -0.5$.

reduced, which leads to a larger step size $\gamma(t_{k+1})$, since γ is a monotone decreasing function. In this way, this method implements an adaptive step size mechanism.

The theoretical convergence properties of this method in one-dimensional optimization problems were studied by [16]. [17] extended the analysis to multidimensional problems. It provides a proof of "almost-sure" convergence and a proof of asymptotical normality. The proof of almost-sure convergence implies that

$$\lim_{k\to\infty} \mu_k = \hat{\mu}, \tag{3.7}$$

"with probability" 1. The proof of asymptotical normality tells us something about the rate of convergence:

$$\sqrt{k}(\mu_k - \hat{\mu}) \to^d N(0, V) \tag{3.8}$$

where $\rightarrow^d$ indicates convergence in distribution and $N(0, V)$ denotes a multivariate normal distribution with mean 0 and covariance matrix V.

The need for setting $a, A, \alpha, f_{MAX}, f_{MIN}, \omega$ complicates the usage of ASGD for image registration, since it is nontrivial to select appropriate parameters for different objective function with different input image data.

3.2 Adaptive Stochastic Conjugate Gradient

From our previous study, we found that there are some drawback for the existing optimization algorithms to solve the 4D nonlinear image registration problem. As we know, the convergence of the deepest gradient descent method is often very slow which will cause large computation. Conjugate gradient converges fast and needs little memory for large-scale problem. But it is poorly scaled for length and requires more iteration in the line search algorithm. The most common Quasi-Newton algorithms are BFGS method and its low-memory extension, LBFGS. BFGS has high space and time complexity. Although LBFGS has better memory management, it still suffers higher computation time in the line search. Another disadvantage for deterministic gradient descent method is that it requires analytic or accurate gradient. Since usually the derivative of the intensity based objective function is approximated by the finite difference method, it is prone to be less accurate in the noisy image. And it is also very expansive to calculate the exact gradient. Compared to other gradient descent based methods, the current adaptive stochastic gradient descent method(ASGD) [5] works with stochastic approximations of the objective function derivatives, and, thus, requires little computation time per iteration. Also its step size adjusts adaptively according to the gradients in the two consecutive steps. However, this method has many free parameters needed user to specify which are non-trivial.

We develop a new algorithm based on the combination of the conjugate gradient and adaptive stochastic gradient method. From our experiments, we observe that this new al-

gorithm converges fast and has few parameters as the conjugate gradient method, and it owns little gradient computation time per iteration and adjusts the step size adaptively as the ASGD method. The main algorithm is that the search direction of ASGD is replaced by stochastic approximations the conjugate gradient of the cost function. Similar to that the conjugate gradient method often converges faster than the gradient descent method, this ASCG method usually converges faster than ASGD. The step size of this algorithm is based on an approximation of the Lipschitz constant of the gradient function. This scheme introduces only one extra parameter to determine the step size and adjust it adaptively.

More specifically, it is defined as:

$$\mu_{k+1} = \mu_k + \gamma(k)\tilde{d}_k, k = 0, 1, \ldots, K. \tag{3.9}$$

Step size $\gamma(k) = \frac{\alpha}{4\times L_k}$, where α is a constant parameters that require user to specify as the input. L_k from the approximation of the Lipschitz constant of the gradient function which has the following formulation:

$$\begin{cases} L_0 = ||\tilde{g}_0||, \\ L_k = \max\{\frac{||\tilde{g}_k - \tilde{g}_{k-1}||}{||\mu_k - \mu_{k-1}||}, L_{k-1}\}, \quad k = 1, \ldots, K. \end{cases} \tag{3.10}$$

The search direction $\tilde{d}_k$ is defined as:

$$\begin{cases} \tilde{d}_0 = \tilde{g}_0, \\ \tilde{d}_k = \tilde{g}_k - \beta_k \tilde{d}_{k-1}, \quad \beta_k = \frac{\tilde{g}_{k+1}^T(\tilde{g}_{k+1} - \tilde{g}_k)}{||\tilde{g}_k||^2}, \quad k = 1, \ldots, K. \end{cases} \tag{3.11}$$

We also have done a comparison experiment with different step size and search direction. It indicates the scheme we chosen gives the best registration performance (see Section4.2).

The convergence properties of ASCG can be analyzed in a similar way of ASGD [17]. Here, we only give our assumption and proposition.

Assumption 1:

1. The stochastic gradient converges to the exact gradient almost surely, i.e.,

$$\tilde{g}_k \to g(\mu_k), \tag{3.12}$$

   with probability 1.

2. The objective function is bounded below.

3. The objective function is Lipschitz continuously differentiable, that is, there exists a constant $L > 0$ such that

$$||g(\mu) - g(\tilde{\mu})|| \leq L||\mu - \tilde{\mu}||, \tag{3.13}$$

   where $g(\mu)$ is the gradient of the objective function.

This assumption implies that there is a constant $\bar{\gamma}$ such that

$$||g(\mu)|| \leq \bar{\gamma}. \tag{3.14}$$

Proposition:

Suppose the Assumption 1 hold, if step size $\lambda_k \in (0, \frac{1}{4L}]$, where L is the Lipschitz constant of the gradient function g, then

$$E[\|g(\mu_k)\|^2] \leq O(\frac{1}{k}). \tag{3.15}$$

Thus this algorithm can get the optimum value when $k \to \infty$. Also in practice, the algorithm is stopped after a specified maximum number of iterations.

4 Experimental Results

In this chapter, we use our proposed optimization algorithm (ASCG) to solve the 4D image registration problem and compare it with other general optimization methods. We adopt linear interpolation in the spatial domain for the derivation of intensity values for any point not on a grid. Our algorithm was implemented in C++ using an computer with Intel Xeon X5570 @2.93GHz, 8GB RAM. Throughout this chapter, we choose the maximum iteration number $K = 2000$.

4.1 Experiment Setup

We perform 4D registration using our algorithm on two public benchmark datasets: POPI [1] and DIR-lab [18]. The dataset from POPI has one 4D CT series including ten 3D volume images ($482 \times 360 \times 141$ pixels) representing ten different phases of one breathing cycle. We also select five datasets from the DIR-lab dataset (Case-1 to Case-5) where landmarks are available. Each dataset contains 6 sequential volume images. We also apply our algorithm into clinical real patient Lung/Tumor data from UT Southwestern Medical Center. The resolution for this dataset is $512 \times 512 \times 152$ pixels $\times 8$ frames. This CT pixel unit can be converted to real physical space unit millimeter by multiplying a scaling factor (recorded in the image header file). Consistent landmarks are also available in the benchmark to measure the accuracy of the registration. Denoting the landmarks on frame-t as $Q_t = \{q_{t,1}, q_{t,2}, \ldots, q_{t,n}\}$, the registration accuracy with respect to frame-r can be measured by a *Mean Target Registration Error* (MTRE) (see Fig.4.1):

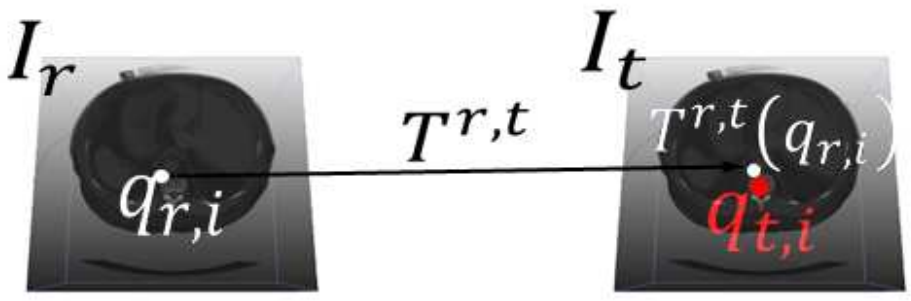


FIGURE 4.1. The illustration of MTRE between two points in the images I_r and I_t .

$$D_r = \frac{1}{n|\Gamma|} \sum_{t \in \Gamma} \sum_{q_{r,i} \in Q_r} ||T^{rt}(q_{r,i}) - q_{t,i}||, \tag{4.1}$$

where T^{rt} is the transformation between frames r and t, composed by the forward and inverse parameterizations following equation (1.2).

4.2 The Choice of the Optimum Scheme for ASCG

In our first experiment, we test different combinations with different step sizes and search directions to get the formulation of ASCG which achieves the best performance. As given in eq 2.2 our iteration scheme is:

$$\mu_{k+1} = \mu_k + \lambda_k d_k, k = 0, 1, 2, \ldots, K. \tag{4.2}$$

We try the following combinations:

1. Step-size: $\lambda_k = \gamma_k = \frac{\alpha}{4L_k}$, search direction: $d_k = \tilde{d}_k$, with $\beta^{PRP} = \frac{\tilde{g}_{k+1}^T(\tilde{g}_{k+1} - \tilde{g}_k)}{||\tilde{g}_k||^2}$,
2. Step-size: $\lambda_k = \gamma_k = \frac{\alpha}{4L_k}$, search direction: $d_k = \tilde{d}_k$, with $\beta^{FR} = \frac{||\tilde{g}_{k+1}^|||^2}{||\tilde{g}_k||^2}$,
3. Step-size: $\lambda_k = \gamma(t_k) = \frac{a}{t_k + A}$, search direction: $d_k = \tilde{d}_k$, with $\beta^{PRP} = \frac{\tilde{g}_{k+1}^T(\tilde{g}_{k+1} - \tilde{g}_k)}{||\tilde{g}_k||^2}$,
4. $\lambda_k = \gamma(t_k) = \frac{a}{t_k + A}$, search direction: $d_k = \tilde{d}_k$, with $\beta^{FR} = \frac{||\tilde{g}_{k+1}^|||^2}{||\tilde{g}_k||^2}$,
5. $\lambda_k = \gamma(t_k) = \frac{a}{t_k + A}$, search direction: $d_k = -\tilde{g}_k$.

TABLE 4.1. The landmark prediction error $D_{1,6}$ and its standard deviation $\sigma_{1,6}$ (in mm) with different search direction and step size. This is the registration between I_1 and I_6 on the POPI data.

$D_{1,6}(\sigma_{1,6})$	Scheme 1 (ASCG)	Scheme 2	Scheme 3	Scheme 4	Scheme 5 (ASGD)
POPI(I_1, I_6)	1.76(0.95)	2.90(1.46)	32.10(32.90)	/	3.59(1.68)

where L_k is defined in the Eq.3.10, $\tilde{d}_k$ is defined in the Eq. 3.11 and t_k is defined in the Eq.3.5. Note that the last scheme is equivalent to the original ASGD method. '/'indicates the algorithm is not converged until the maximum iterative step $K = 2000$ reached.

We apply these schemes into the 4D registration on the public available POPI Lung CT data [1]. The total unknown parameters is 93000. Table4.1 shows the registration accuracy between I_1 and I_6 which corresponding to the End of Expiration and End of Inspiration status. The error between these two states indicates the maximum registration error during the respiratory cycles.

$$D_{r,t} = \frac{1}{n} \sum_{q_{r,i} \in Q_r} ||T^{rt}(q_{r,i}) - q_{t,i}||, \tag{4.3}$$

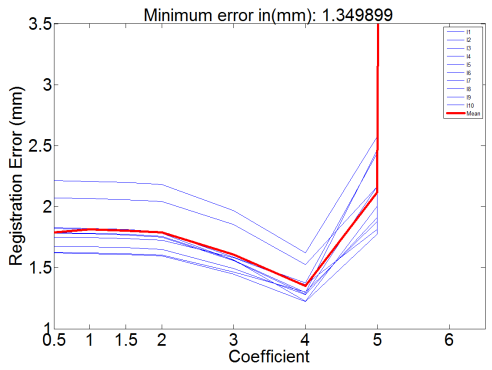
We can see that the scheme 1 achieves the best registration accuracy. Thus in our following experiments, we will adopt the step-size and search direction as scheme 1.

4.3 ASCG with Different Image Resolution

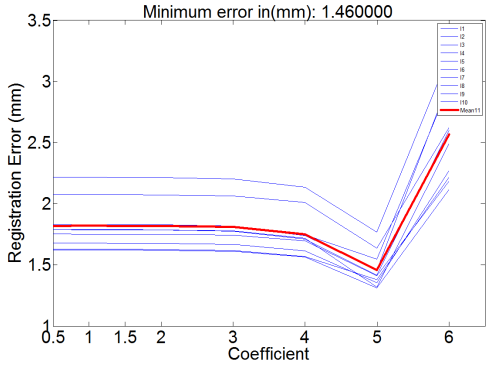
In this section, we investigate the influence of the image resolution on the choice of α which determines the step-size in each iteration. Registration experiments were performed on the POPI CT lung data Given the 4D CT images, we first smooth the image, then do the resampling with different resolution. Let (X, Y, Z) be the original image size, then the resolution N indicates the smoothed and resampled image of which size is $(\frac{X}{N}, \frac{Y}{N}, \frac{Z}{N})$ The total unknown parameters for resolution 8 is 29568, resolution 4 is 135520, resolution 2 is 707200 and resolution 1 is 4712000.

Fig.4.2 shows the registration accuracy with respect to the different coefficient α. The blue line shows the registration error of the individual images. The red shows their mean value. We can see that the image with higher resolution is more robust to the step-size. For example, in Fig.4.2(a) when resolution is 8, it is not converged when $\alpha > 5$, when resolution is 4, the error increase largely when $\alpha > 6$, when resolution is 2, it is converged, but also has a sharp decrease from $\alpha = 3$ to $\alpha = 4$, and when resolution is 1, the registration accuracy does not change much to the step-size. However, the best registration accuracy is achieved when resolution is 4 with $\alpha = 5$.

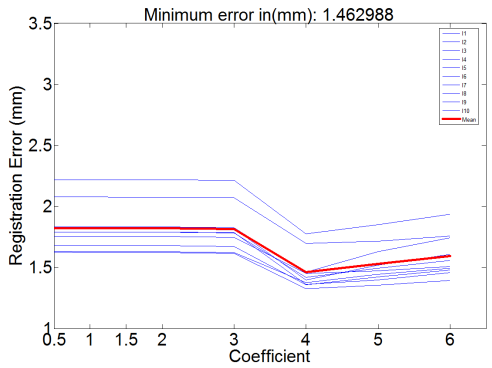
Fig.4.3 shows the computation time in different image resolution. As we can see the image with lower resolution has less time complexity. Thus, in our later experiments we adopt the image in resolution 4 with $\alpha = 5$ to achieve better registration accuracy and efficiency.



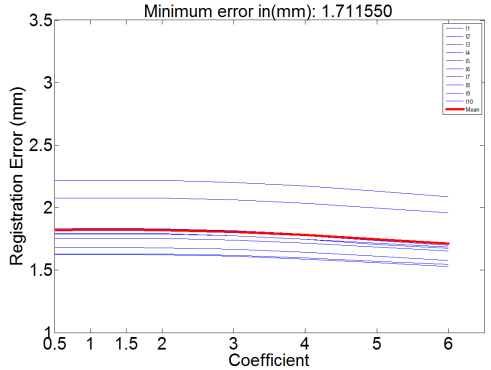
(a) Resolution 8



(b) Resolution 4



(c) Resolution 2



(d) Resolution 1

FIGURE 4.2. The registration accuracy w.r.t. the coefficient α under different image resolution.

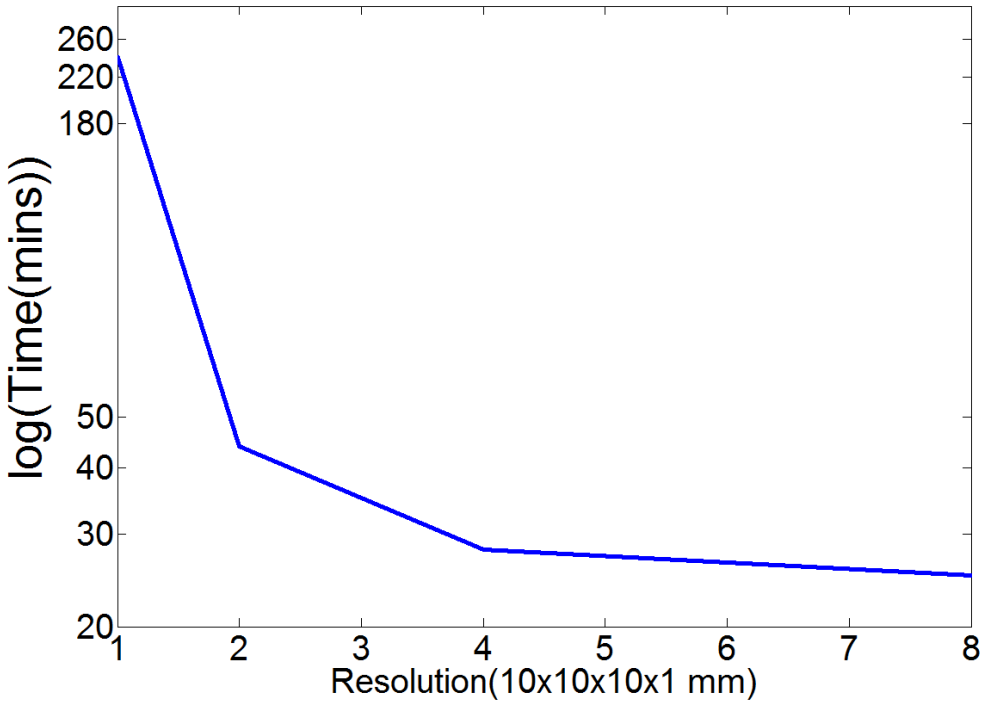
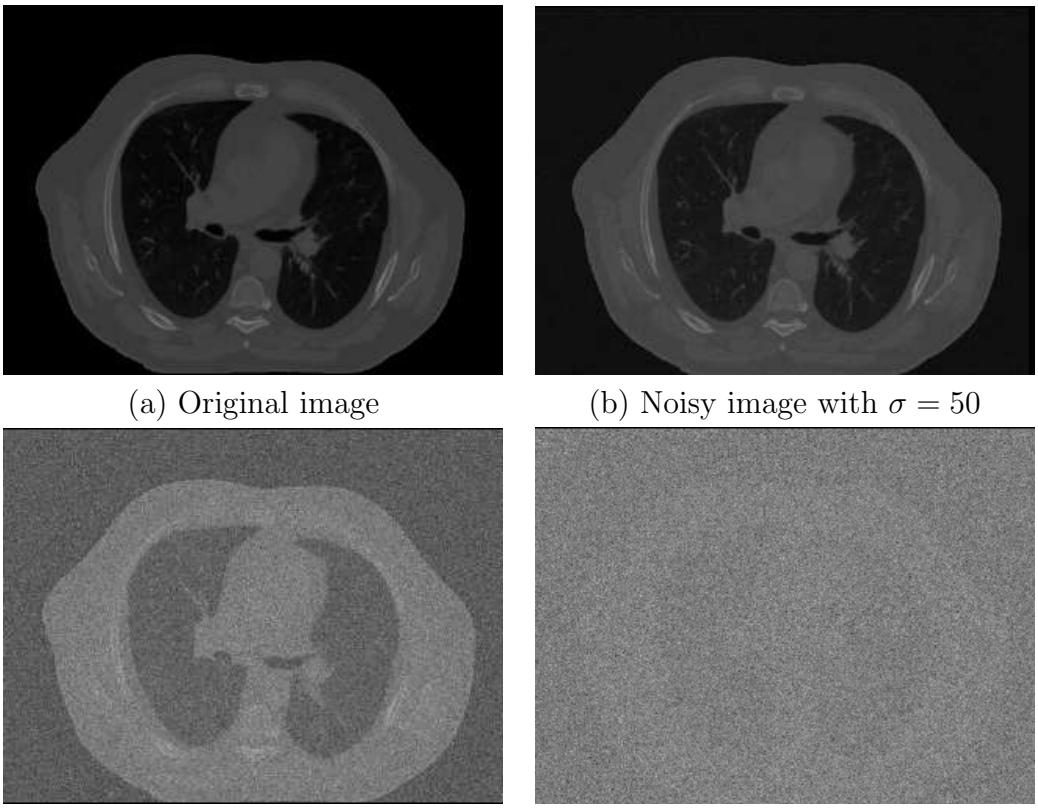


FIGURE 4.3. Time comparison with different image resolution.

4.4 ASCG with Image Noise

In this section, we will show that our algorithm is also robust to image noise. We add the gaussian noise into the POPI image data with the mean equal to 0 and the variance from 0 to 4000. Fig.4.4 shows the 2D cross section of the samples images. Fig.4.5 shows the registration accuracy with respect to the image noise under different optimization schemes. We can see that the Quasi-Newton (LBFGS) method is quite sensitive to the image noise since it requires accurate gradient computation. But the gradient becomes less accurate as the noise of the image increases. Also we can find the registration error of the steepest gradient descent method becomes high as the increasing of image noises. The conjugate gradient method, ASGD and our proposed method ASCG are robust to the image noise while ASCG achieves the best registration accuracy.



(a) Original image (b) Noisy image with $\sigma = 50$

(c) Noisy image with $\sigma = 500$ (d) Noisy image with $\sigma = 4000$

FIGURE 4.4. The 2D cross section of the 4D images.

4.5 Optimization Comparison

In this section, we will compare our proposed optimization scheme (ASCG) with other general optimization methods which include steepest Gradient Descent (GD) method, Conjugate Gradient(CG) method, LBFGS method, and ASGD method. The total unknown parameters for POPI is 135520, for DIR-Lab1 is 41472, for DIR-Lab2/3/4/5 is 25920.

Table 4.2 shows our comparison on the POPI dataset. We can see our ASCG algorithm achieves 22% higher accuracy than the general optimization methods. Table 4.3 shows our algorithm also performs better than the general optimization methods on the DIR-Lab datasets.

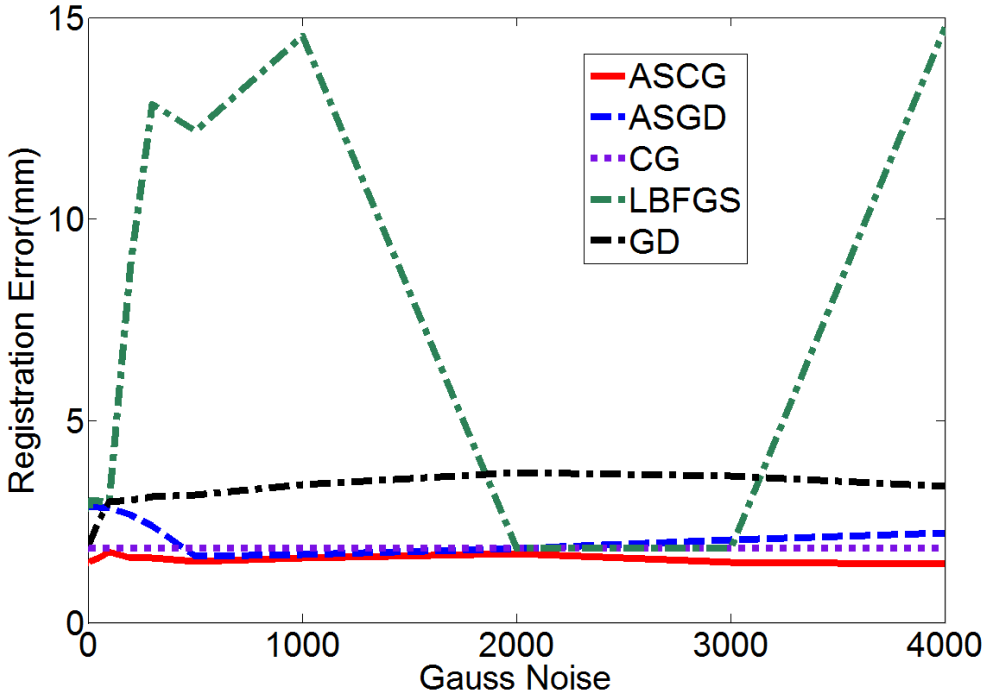


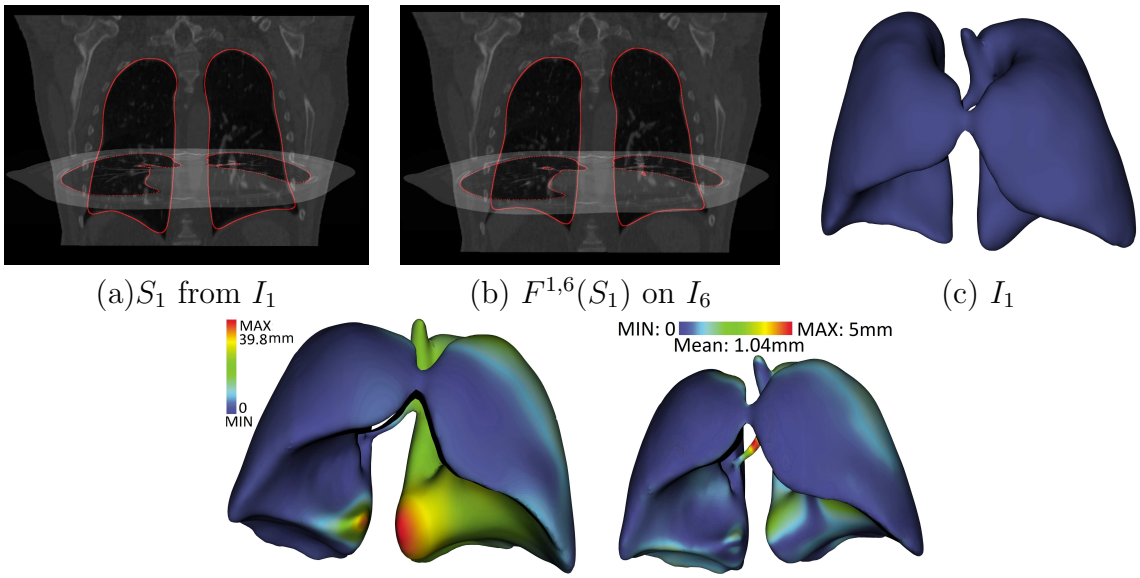
FIGURE 4.5. Registration accuracy with respect to the image noise under different optimization schemes.

TABLE 4.2. The landmark predication error D_i and its standard deviation σ_i (in mm) of i^{th} time frame on the POPI-data [1] with different optimization methods. $\bar{D}$ is the average MTRE.

	$D_1(\sigma_1)$	$D_2(\sigma_2)$	$D_3(\sigma_3)$	$D_4(\sigma_4)$	$D_5(\sigma_5)$	$D_6(\sigma_6)$	$D_7(\sigma_7)$	$D_8(\sigma_8)$	$D_9(\sigma_9)$	$D_0(\sigma_0)$	$\bar{D}$
GD	2.2(2.0)	2.1(1.8)	1.9(1.5)	2.0(1.5)	2.2(1.5)	2.6(2.2)	2.3(1.8)	1.9(1.3)	1.9(1.5)	2.1(1.9)	2.1
CG	1.8(1.6)	1.7(1.5)	1.6(1.2)	1.7(1.3)	2.0(1.5)	2.2(1.6)	1.8(1.3)	1.6(1.1)	1.6(1.2)	1.7(1.5)	1.8
LBFGS	2.5(2.0)	2.3(2.0)	2.1(1.6)	2.0(1.3)	2.3(1.8)	2.9(2.3)	2.8(2.2)	2.1(1.4)	1.9(1.4)	2.3(1.9)	2.3
ASGD	3.3(2.5)	2.8(2.3)	2.4(1.7)	2.4(1.7)	2.8(2.2)	3.5(2.6)	3.2(2.5)	2.5(2.7)	2.3(1.5)	2.8(2.2)	2.8
ASCG	1.4(1.3)	1.4(1.3)	1.3(1.0)	1.4(1.0)	1.6(1.2)	1.7(1.3)	1.5(1.1)	1.3(0.9)	1.3(1.0)	1.3(1.1)	1.4

TABLE 4.3. The landmark prediction error D and its standard deviation σ (in mm) on the DIR-LAB data set with different optimization methods.

$D(\sigma)$	GD	CG	LBFGS	ASGD	ASCG
DIR-Lab1	1.97(1.18)	2.21(1.03)	2.21(1.08)	2.23(1.02)	1.97(1.18)
DIR-Lab2	1.24(0.77)	1.42(0.91)	1.28(0.74)	1.24(0.77)	1.22(0.74)
DIR-Lab3	2.16(1.01)	2.18(1.01)	2.14(1.09)	3.80(1.96)	2.04(0.98)
DIR-Lab4	1.87(1.00)	4.42(1.85)	1.81(1.06)	2.11(1.13)	1.77(1.08)
DIR-Lab5	2.42(1.68)	3.67(2.28)	2.42(1.66)	2.35(1.69)	2.34(1.58)



(a)S_1 from I_1 (b) $F^{1,6}(S_1)$ on I_6 (c) I_1

(d) $d(F^{1,6}(I_1), I_1)$ (e) $H(F^{1,6}(I_1), I_6)$

FIGURE 4.6. Lung/Tumor Tracking via a Deforming Surface Geometry. (a, b) show the alignment of iso-contours and the scanned images. (d) shows the color-coded displacement field of $T^{1,6}(I_1)$ from I_1 in (c); (e) visualizes the Hausdorff distance from the deformable model to the scan.

4.6 Application: Motion Modeling of Clinical Lung Tumor Scans

Our last experiment is apply to the 4D image registration with ASCG optimization into clinic real patient data. We use our results to describe the lung/tumor deformation from clinic CT scans. We first perform image segmentation and construct finite element mesh models [19], then use the 4D mapping to compute its deformation. Fig. 4.6 [20] illustrates a few snapshots of this tracking. (a) shows the surface contour segmented from frame-1 and (b) shows the deformed contour on frame-6; (c) and (d) show the color-encoded displacement fields of our deformable model; (e) illustrates the matching error measured by Hausdorff distance. This illustrated matching, between the maximum inhalation (I_6) and maximum exhalation status (I_1) which undergoes a largest deformation, infers the maximum matching errors during the respiratory cycles.

5 Conclusion

In this thesis, we develop an Adaptive Stochastic Conjugate Gradient (ASCG) algorithm to solve the optimization in temporal medical image registration. Objective functions in these optimization problems are usually complex and have large number of variables. Our proposed ASCG algorithm effectively combines the conjugate gradient and adaptive stochastic gradient method. This new algorithm converges fast and has few parameters like the conjugate gradient method. During each iteration, its gradient computation is efficient and the step size adaptively adjustable.

Compared to the previous stochastic gradient approximations, this algorithm converges faster, which is similar to the fact that the convergence of conjugate gradient is faster than the deepest descent method. The step size of this algorithm approximates the Lipschitz constant of the stochastic gradient function. This scheme introduces only one extra parameter to determine the step size.

Our preliminary results demonstrate its efficiency on the public available 4D Lung CT data and our clinical Lung/Tumor CT data using the general 4D image registration model. The results indicate that our ASCG algorithm achieves 22% higher accuracy on the POPI data and consistently performs better than the existing methods on other data sets (DIR-Lab dataset and our clinical dataset). Furthermore, we also demonstrate that compared with other methods, our ASCG algorithm is more robust to image noise.

References

[1] J. Vandemeulebroucke, D. Sarrut, and P. Clarysse. Point-validated pixel-based breathing thorax model. In *Intern. Conf. on the Use of Computers in Radiation Therapy, 2007*.

[2] H. Xu, P. Chen, W. Yu, A. Sawasnt, S. Iyengar, and X. Li. Feature-aligned 4D Spatiotemporal Image Registration. In *Intern. Conf. on Pattern Recognition*, pages 2639–2642, 2012.

[3] G. Christensen and H. Johnson. Consistent image registration. *Medical Imaging, IEEE Transaction on*, 20(7):568 –582, 2001.

[4] Y.-H. Dai. A family of hybrid conjugate gradient methods for unconstrained optimization. *Mathematics of Computation*, 72(243):1317–1328, 2003.

[5] S. Klein, J. P. Pluim, M. Staring, and M.A. Viergever. Adaptive stochastic gradient descent optimisation for image registration. *Int. J. Comput. Vision*, 81(3):227–239, March 2009.

[6] J.Nocedal and S.J.Wright. *Numerical Optimization*. Springer-Verlag, New York, 2006.

[7] W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization*, 16(1):170–192, 2005.

[8] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1):35–58, 2006.

[9] J.E.Dennis, Jr., and J.J.More. Quasi-newton methods, motivation and theory. *SIAM Review,*, 19(1):46 – 89, 1977.

[10] J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

[11] S. Klein, M. Staring, and J. P W Pluim. Evaluation of optimization methods for nonrigid medical image registration using mutual information and b-splines. *Image Processing, IEEE Transactions on*, 16(12):2879–2890, 2007.

[12] Jorge J. Moré and David J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Softw.*, 20(3):286–307, 1994.

[13] D.P.Bertsekas. *Nonlinear Programming*. Athena Scientific, Massachusetts, 1999.

[14] H. Mukai. Readily implementable conjugate gradient methods. *Mathematical Programming*, 17(1):298–319, 1979.

[15] H.J.Kushner and G.G.Yin. *Stochastic Approximation and Recursive Algorithms and Applications (2nd Edition)*. Springer-Verlag, New York, 2003.

[16] A. Plakhov and P. Cruz. A stochastic approximation algorithm with step-size adaptation. *Journal of Mathematical Sciences*, 120(1):964–973, 2004.

[17] P.Cruz. *Almost sure convergence and asymptotic normality of a generalization of kesten's stochastic approximation algorithm for multidimensional case*. Cadernos de Matematics, Serie de Investigacao, Collection of University of Aveiro, Department of Mathematics, (Technical Report), 2005.

[18] R. Castillo, E. Castillo, R. Guerra, V. E. Johnson, T. McPhail, A.K. Garg, and T.Guerrero. A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets. *Phys Med Biol*, 54:1849–1870, 2009.

[19] S. S. Iyengar, X. Li, H. Xu, S. Mukhopadhyay, N. Balakrishnan, A. Sawant, and P. Iyengar. Toward more precise radiotherapy treatment of lung tumors. *Computer*, 45:59–65, 2012.

[20] H. Xu and X. Li. A symmetric 4d registration algorithm for respiratory motion modeling. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2013.

Vita

Huanhuan Xu was born in Jiangxi, China, in 1984. She graduated from Centra China Normal University, in 2006 with double bachelors' degree in the Computer Science Department and the Mathematics Department. She earned a master degree in the Automation Department from University of Science and Technology of China in 2009. She got her Ph.D. degree in the Department of Electrical and Computer Engineering in August 2013. Currently she is a candidate for the degree of Master of Science in mathematics with concentration in applications, which will be awarded in Dec. 2013.

She was offered a graduate research assistantship from Center for Computation and Technology and a Mark and Carolyn Guidry doctoral fellowship from Department of Electrical and Computer Engineering in 2009-2013. Her research interests include Nonlinear Programming and its Applications, Medical Image Processing and Analysis, Computer Graphics and Geometric Modeling.