Fast Automatic Step Size Estimation for Gradient Descent Optimization of Image Registration

Yuchuan Qiao*, Baldur van Lew, Boudewijn P. F. Lelieveldt, and Marius Staring

Abstract—**Fast automatic image registration is an important prerequisite for image-guided clinical procedures. However, due to the large number of voxels in an image and the complexity of registration algorithms, this process is often very slow. Stochastic gradient descent is a powerful method to iteratively solve the registration problem, but relies for convergence on a proper selection of the optimization step size. This selection is difficult to perform manually, since it depends on the input data, similarity measure and transformation model. The Adaptive Stochastic Gradient Descent (ASGD) method is an automatic approach, but it comes at a high computational cost. In this paper, we propose a new computationally efficient method (fast ASGD) to automatically determine the step size for gradient descent methods, by considering the observed distribution of the voxel displacements between iterations. A relation between the step size and the expectation and variance of the observed distribution is derived. While ASGD has quadratic complexity with respect to the transformation parameters, fast ASGD only has linear complexity. Extensive validation has been performed on different datasets with different modalities, inter/intra subjects, different similarity measures and transformation models. For all experiments, we obtained similar accuracy as ASGD. Moreover, the estimation time of fast ASGD is reduced to a very small value, from 40 s to less than 1 s when the number of parameters is 105, almost 40 times faster. Depending on the registration settings, the total registration time is reduced by a factor of 2.5–7 × for the experiments in this paper.**

Index Terms—**(Stochastic) gradient descent, gradient descent optimization, image registration, optimization step size.**

I. Introduction

IMAGE registration aims to align two or more images and is an important technique in the field of medical image analysis. It has been used in clinical procedures including radiotherapy and image-guide surgery, and other general image analysis tasks, such as automatic segmentation [1]–[4]. However, due to the large number of image voxels, the large amount of transformation parameters and general algorithm complexity, this process is often very slow [5]. This renders the technique impractical in time-critical clinical situations, such as intra-operative procedures.

To accelerate image registration, multiple methods have been developed targeting the transformation model, the interpolation scheme or the optimizer. Several studies investigate the use of state-of-the-art processing techniques exploiting multi-threading on the CPU or also the GPU [6], [7]. Others focus on the optimization scheme that is used for solving image registration problems [8]–[10]. Methods include gradient descent [11], [12], Levenberg-Marquardt [13], [14], quasi-Newton [15], [16], conjugate gradient descent [10], evolution strategies [17], particle swarm methods [18], [19], and stochastic gradient descent methods [20], [21]. Among these schemes, the stochastic gradient descent method is a powerful method for large scale optimization problems and has a superb performance in terms of computation time, with similar accuracy as deterministic first order methods [10]. Deterministic second order methods gave slightly better accuracy in that study, but at heavily increased computational cost. It may therefore be considered for cases where a high level of accuracy is required, in a setting where real-time performance is not needed.

In this study, we build on the stochastic gradient descent technique to solve the optimization problem of image registration [12]:

$$\widehat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}} \mathcal{C}(I_F, I_M \circ \boldsymbol{T}_{\boldsymbol{\mu}}), \tag{1}$$

in which $I_F(\boldsymbol{x})$ is the d-dimensional fixed image, $I_M(\boldsymbol{x})$ is the d-dimensional moving image, $\boldsymbol{T}(\boldsymbol{x}, \boldsymbol{\mu})$ is a parameterized coordinate transformation, and $\mathcal{C}$ the cost function to measure the dissimilarity between the fixed and moving image. To solve this problem, the stochastic gradient descent method adopts iterative updates to obtain the optimal parameters using the following form:

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k - \gamma_k \tilde{\boldsymbol{g}}_k, \tag{2}$$

where k is the iteration number, γ_k the step size at iteration k, $\tilde{\boldsymbol{g}}_k = \boldsymbol{g}_k + \boldsymbol{\epsilon}_k$ the stochastic gradient of the cost function, with the true gradient $\boldsymbol{g}_k = \partial\mathcal{C}/\partial\boldsymbol{\mu}_k$ and the approximation error $\boldsymbol{\epsilon}_k$. The stochastic gradient can be efficiently calculated using a subset of voxels from the fixed image [21] or using simultaneous perturbation approximation [22]. As shown previously [10], stochastic gradient descent has superior performance in terms of computation time compared to deterministic gradient descent and deterministic second order methods such as quasi-Newton, although the latter frequently obtains somewhat lower objective values. Similar to second order methods, stochastic gradient descent is less prone to get stuck in small

Manuscript received August 07, 2015; accepted August 30, 2015. Date of publication September 03, 2015; date of current version February 01, 2016. This research was supported by the Netherlands Organization for Scientific Research (NWO VENI 639.021.919), by the Dutch national e-infrastructure with the support of the SURF Foundation (e-infra140085), and by the China Scholarship Council (201206130066). *Asterisk indicates corresponding author.*

*Y. Qiao is with the Division of Image Processing, Department of Radiology, Leiden University Medical Center, 2300RC Leiden, The Netherlands.

B. van Lew, B. P. F. Lelieveldt, and M. Staring are with the Division of Image Processing, Department of Radiology, Leiden University Medical Center, 2300RC Leiden, The Netherlands.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2015.2476354

local minima compared to deterministic gradient descent [23], [24]. Almost-sure convergence of the stochastic gradient descent method is guaranteed (meaning that it will converge to the local minimum ”with probability 1”), provided that the step size sequence is a non-increasing and non-zero sequence with $\sum_{k=1}^{\infty} \gamma_k = \infty$ and $\sum_{k=1}^{\infty} \gamma_k^2 < \infty$ [25]. A suitable step size sequence is very important, because a poorly chosen step size will cause problems of estimated value ”bouncing” if this step size is too large, or slow convergence if it is too small [26], [27]. Therefore, an exact and automatically estimated step size, independent of problem settings, is essential for the gradient-based optimization of image registration. Note that for deterministic quasi-Newton methods the step size is commonly chosen using an (in)exact line search.

Methods that aim to solve the problem of step size estimation can be categorized in three groups: manual, semi-automatic, and automatic methods. In 1952, Robbins and Monro [25] proposed to manually select a suitable step size sequence. Several methods were proposed afterwards to improve the convergence of the Robbins-Monro method, which focused on the construction of the step size sequence, but still required manual selection of the initial step size. Examples include Kesten's rule [28], Gaivoronski's rule [29], and the adaptive two-point step size gradient method [30]. An overview of these methods can be found here [31], [32]. These manual selection methods, however, are difficult to use in the practice, because different applications require different settings. Especially for image registration, different fixed or moving images, different similarity measures or transformation models require a different step size. For example, it has been reported that the step size can differ several orders of magnitude between cost functions [21]. Moreover, manual selection is time-consuming.

Spall [22] used a step size following a rule-of-thumb that the step size times the magnitude of the gradient is approximately equal to the smallest desired change of $\boldsymbol{\mu}$ in the early iterations. The estimation is based on a preliminary registration, after which the step size is manually estimated and used in subsequent registrations. This manual procedure is not adaptive to the specific images, depends on the parameterization $\boldsymbol{\mu}$, and requires setting an nonintuitive ‘desired change’ in $\boldsymbol{\mu}$.

For the semi-automatic selection, Suri [26] and Brennan [27] proposed to use a step size with the same scale as the magnitude of $\boldsymbol{\mu}$ observed in the first few iterations of a preliminary simulation experiment, in which a latent difference of the step size between the preliminary experiment and the current one is inevitable. Bhagalia also used a training method to estimate the step size of stochastic gradient descent optimization for image registration [33]. First, a pseudo ground truth was obtained using deterministic gradient descent. Then, after several attempts, the optimal step size was chosen to find the optimal warp estimates which had the smallest error values compared with the pseudo ground truth warp obtained in the first step. This method is complex and time-consuming as it requires training data, and moreover generalizes training results to new cases.

The Adaptive Stochastic Gradient Descent method (ASGD) [21] proposed by Klein *et al.* automatically estimates the step size. ASGD estimates the distribution of the gradients and the distribution of voxel displacements, and finally calculates the initial step size based on the voxel displacements. This method works for few parameters within reasonable time, but for a large number of transformation parameters, i.e., in the order of 10^5 or higher, the run time is unacceptable and the time used in estimating the step size will dominate the optimization [34]. This disqualifies ASGD for real-time image registration tasks.

In this paper, we propose a new computationally efficient method, fast ASGD (hereafter FASGD), to automatically select the optimization step size for gradient descent optimization, by deriving a relation with the observed voxel displacement. This paper extends a conference paper [34] with detailed methodology and extensive validation, using many different datasets of different modality and anatomical structure. Furthermore, we have developed tools to perform extensive validation of our method by interfacing with a large international computing facility. In Section II, the method to calculate the step size is introduced. The dataset description is given in Section III. The experimental setup to evaluate the performance of the new method is presented in Section IV. In Section V, the experimental results are given. Finally, Sections VI and VII conclude the paper.

II. Method

A commonly used choice for the step size estimation in gradient descent is to use a monotonically non-increasing sequence. In this paper we use the following decaying function, which can adaptively tune the step size according to the direction and magnitude of consecutive gradients, and has been used frequently in the stochastic optimization literature [5], [20], [21], [25], [29], [31], [32], [35], [36]:

$$\gamma_k = \frac{a}{(A + t_k)^{\alpha}}, \tag{3}$$

with $a > 0$, $A \geq 1$, $0 < \alpha \leq 1$, where $\alpha = 1$ gives a theoretically optimal rate of convergence [35], and is used throughout this paper. The iteration number is denoted by k, and $t_k = \max(0, t_{k-1} + f(-\tilde{\boldsymbol{g}}_{k-1}^T \tilde{\boldsymbol{g}}_{k-2}))$. The function f is a sigmoid function with $f(0) = 0$:

$$f(x) = \frac{f_{\max} - f_{\min}}{1 - \left(\frac{f_{\max}}{f_{\min}}\right) e^{-x/\omega}} + f_{\min}, \tag{4}$$

in which $f_{\max}$ determines the maximum gain at each iteration, $f_{\min}$ determines the maximal step backward in time, and ω affects the shape of the sigmoid function [21]. A reasonable choice for the maximum of the sigmoid function is $f_{\max} = 1$, which implies that the maximum step forward in time equals that of the Robbins-Monro method [21]. It has been proven that convergence is guaranteed as long as $t_k \geq 0$ [21], [36]. Specifically, from Assumption A4 [36] and Assumption B5 [21], asymptotic normality and convergence can be assured when $f_{\max} > -f_{\min}$ and $\omega > 0$. In [21, (Equation (59))] $\omega = \zeta \sqrt{Var(\boldsymbol{\varepsilon}_k^T \boldsymbol{\varepsilon}_{k-1})}$ was used, which requires the estimation of the distribution of the approximation error for the gradients, which is time consuming. Moreover, a parameter ζ is introduced which was empirically set to 10%. Setting $\omega = 10^{-8}$ avoids a costly computation, and still guarantees the conditions required for convergence. For the minimum of the sigmoid function we choose $f_{\min} = -0.8$ in this paper, fulfilling the convergence criteria.

In the step size sequence $\{\gamma_k\}$, all parameters need to be selected before the optimization procedure. The parameter α controls the decay rate; the theoretically optimal value is 1 [21], [37]. The parameter A provides a starting point, which has most influence at the beginning of the optimization. From experience [21], [37], $A = 20$ provides a reasonable value for most situations. The parameter a in the numerator determines the overall scale of the step size sequence, which is important but difficult to select, since it is dependent on I_F, I_M, $\mathcal{C}$ and $\boldsymbol{T_\mu}$. The step size can differ substantially between resolutions ([21, Figure 4]) and for different cost functions ([21, Table 2]). This means that the problem of estimating the step size sequence is mainly determined by a. In this work, we therefore focus on automatically selecting the parameter a in a less time-consuming manner.

A. Maximum Voxel Displacement

The intuition of the proposed step size selection method is that the voxel displacements should start with a reasonable value and gradually diminish to zero. The incremental displacement of a voxel $\boldsymbol{x}_j$ in a fixed image domain Ω_F between iteration k and $k+1$ for an iterative optimization scheme is defined as

$$\boldsymbol{d}_k(\boldsymbol{x}_j) = \boldsymbol{T}(\boldsymbol{x}_j, \boldsymbol{\mu}_{k+1}) - \boldsymbol{T}(\boldsymbol{x}_j, \boldsymbol{\mu}_k), \ \forall \boldsymbol{x}_j \in \Omega_F. \tag{5}$$

To ensure that the incremental displacement between each iteration is neither too big nor too small, we need to constrain the voxel's incremental displacement $\boldsymbol{d}_k$ into a reasonable range. We assume that the magnitude of the voxel's incremental displacement $\boldsymbol{d}_k$ follows some distribution, which has expectation $E\|\boldsymbol{d}_k\|$ and variance $Var\|\boldsymbol{d}_k\|$, in which $\|\cdot\|$ is the ℓ^2 norm. For a translation transform, the voxel displacements are all equal, so the variance is zero; for non-rigid registration, the voxel displacements vary spatially, so the variance is larger than zero. To calculate the magnitude of the incremental displacement $\|\boldsymbol{d}_k\|$, we use the first-order Taylor expansion to make an approximation of $\boldsymbol{d}_k$ around $\boldsymbol{\mu}_k$:

$$\boldsymbol{d}_k \approx \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{\mu}}(\boldsymbol{x}_j, \boldsymbol{\mu}_k) \cdot (\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k) = \boldsymbol{J}_j(\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k), \tag{6}$$

in which $\boldsymbol{J}_j = (\partial \boldsymbol{T}/\partial \boldsymbol{\mu})(\boldsymbol{x}_j, \boldsymbol{\mu}_k)$ is the Jacobian matrix of size $d \times |\boldsymbol{\mu}|$. Defining $\boldsymbol{M}_k(\boldsymbol{x}_j) = \boldsymbol{J}(\boldsymbol{x}_j)\boldsymbol{g}_k$ and combining with the update rule $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k - \gamma_k \boldsymbol{g}_k$, $\boldsymbol{d}_k$ can be rewritten as:

$$\boldsymbol{d}_k(\boldsymbol{x}_j) \approx -\gamma_k \boldsymbol{J}(\boldsymbol{x}_j)\boldsymbol{g}_k = -\gamma_k \boldsymbol{M}_k(\boldsymbol{x}_j). \tag{7}$$

For a maximum allowed voxel displacement, Klein [21] introduced a user-defined parameter δ, which has a physical meaning with the same unit as the image dimensions, usually in mm. This implies that the maximum voxel displacement for each voxel between two iterations should be not larger than δ: i.e $\|\boldsymbol{d}_k(\boldsymbol{x}_j)\| \leq \delta, \forall \boldsymbol{x}_j \in \Omega_F$. We can use a weakened form for this assumption:

$$P(\|\boldsymbol{d}_k(\boldsymbol{x}_j)\| > \delta) < \rho, \tag{8}$$

where ρ is a small probability value often 0.05. According to the Vysochanskij Petunin inequality [38], for a random variable X with unimodal distribution, mean μ and finite, non-zero variance σ^2, if $\lambda > \sqrt{(8/3)}$, the following theorem holds:

$$P(|X - \mu| \geq \lambda\sigma) \leq \frac{4}{9\lambda^2}. \tag{9}$$

This can be rewritten as:

$$P(\mu - 2\sigma \leq x \leq \mu + 2\sigma) \approx 0.95. \tag{10}$$

Based on this boundary, we can approximate (8) with the following expression:

$$E\,\|\boldsymbol{d}_k(\boldsymbol{x}_j)\| + 2\sqrt{Var\,\|\boldsymbol{d}_k(\boldsymbol{x}_j)\|} \leq \delta. \tag{11}$$

This is slightly different from the squares used in [21, Equation (42)], which avoids taking square roots for performance reasons. In this paper we are interested in the incremental displacements, not its square. Combining with (7), we obtain the relationship between step size and maximum voxel displacement as follows:

$$\gamma_k \left(E\,\|\boldsymbol{M}_k(\boldsymbol{x}_j)\| + 2\sqrt{Var\,\|\boldsymbol{M}_k(\boldsymbol{x}_j)\|} \right) \leq \delta. \tag{12}$$

B. Maximum Step Size for Deterministic Gradient Descent

From the step size function $\gamma(k) = a/(k+A)^\alpha$, it is easy to find the maximum step size $\gamma_{\max} = \gamma(0) = a/A^\alpha$, and the maximum value of a, $a_{\max} = \gamma_{\max} A^\alpha$. This means that the largest step size is taken at the beginning of the optimization procedure for each resolution. Using (12), we obtain the following equation of $a_{\max}$:

$$a_{\max} = \frac{\delta A^\alpha}{E\|\boldsymbol{M}_0(\boldsymbol{x}_j)\| + 2\sqrt{Var\|\boldsymbol{M}_0(\boldsymbol{x}_j)\|}}. \tag{13}$$

For a given δ, the value of a can be estimated from the initial distribution of $\boldsymbol{M}_0$ at the beginning of each resolution.

C. Noise Compensation for Stochastic Gradient Descent

The stochastic gradient descent method combines fast convergence with a reasonable accuracy [10]. Fast estimates of the gradient are obtained using a small subset of the fixed image voxels, randomly chosen in each iteration. This procedure introduces noise to the gradient estimate, thereby influencing the convergence rate. This in turn means that the optimal step size for *stochastic* gradient descent will be different compared to *deterministic* gradient descent. When the approximation error $\boldsymbol{\epsilon} = \boldsymbol{g} - \tilde{\boldsymbol{g}}$ increases, the search direction $\tilde{\boldsymbol{g}}$ is more unpredictable, thus a smaller and more careful step size is required. Similar to [21] we assume that $\boldsymbol{\epsilon}$ is a zero mean Gaussian variable with small variance, and we adopt the ratio between the expectation of the exact and approximated gradient to modify the step size $a_{\max}$ as follows:

$$\eta = \frac{E\|\boldsymbol{g}\|^2}{E\|\tilde{\boldsymbol{g}}\|^2} = \frac{E\|\boldsymbol{g}\|^2}{E\|\boldsymbol{g}\|^2 + E\|\boldsymbol{\epsilon}\|^2}. \tag{14}$$

D. Summary and Implementation Details

1) The Calculation of $a_{\max}$ for Exact Gradient Descent: The cost function used in voxel-based image registration usually takes the following form:

$$\boldsymbol{C}(\boldsymbol{\mu}) = \frac{1}{|\Omega_F|} \sum_{\boldsymbol{x}_j \in \Omega_F} \Psi\left(I_F(\boldsymbol{x}_j), I_M(\boldsymbol{T}(\boldsymbol{x}_j, \boldsymbol{\mu}))\right), \tag{15}$$

in which Ψ is a similarity measure, Ω_F is a discrete set of voxel coordinates from the fixed image and $|\Omega_F|$ is the cardinality of this set. The gradient $\boldsymbol{g}$ of this cost function is:

$$\boldsymbol{g} = \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{\mu}} = \frac{1}{|\Omega_F|} \sum_{\boldsymbol{x}_j \in \Omega_F} \frac{\partial \boldsymbol{T}'}{\partial \boldsymbol{\mu}} \frac{\partial I_M}{\partial \boldsymbol{x}} \frac{\partial \Psi}{\partial I_M}. \tag{16}$$

The reliable estimate of $a_{\max}$ relies on the calculation of the exact gradient. We obtain a trade-off between the accuracy of computing $\boldsymbol{g}$ with its computation time, by randomly selecting a sufficiently large number of samples from the fixed image. Specifically, to compute (16) we use a subset $\Omega_F^1 \subset \Omega_F$ of size N_1 equal to the number of transformation parameters $P = |\boldsymbol{\mu}|$.

Then, $\boldsymbol{J}_j = (\partial \boldsymbol{T}/\partial \boldsymbol{\mu})(\boldsymbol{x}_j, \boldsymbol{\mu}_k)$ is computed at each voxel coordinate $\boldsymbol{x}_j \in \Omega_F^1$. The expectation and variance of $\|\boldsymbol{M}_0(\boldsymbol{x}_j)\|$ can be calculated using the following expressions:

$$E\|\boldsymbol{M}_0(\boldsymbol{x}_j)\| = \frac{1}{N_1} \sum_{\boldsymbol{x}_j \in \Omega_F^1} \|\boldsymbol{M}_0(\boldsymbol{x}_j)\|, \tag{17}$$

$$Var\|\boldsymbol{M}_0(\boldsymbol{x}_j)\| = \frac{1}{N_1 - 1} \sum_{\boldsymbol{x}_j \in \Omega_F^1} \left(\|\boldsymbol{M}_0(\boldsymbol{x}_j)\| - E\|\boldsymbol{M}_0(\boldsymbol{x}_j)\|\right)^2. \tag{18}$$

2) The Calculation of η: The above analysis reveals that the noise compensation factor η also influences the initial step size. This factor requires computation of the exact gradient $\boldsymbol{g}$ and the approximate gradient $\tilde{\boldsymbol{g}}$. Because the computation of the exact gradient using all voxels is too slow, uniform sampling is used, where the number of samples is determined empirically as $N_2 = \min(100000, |\Omega_F|)$. To obtain the stochastic gradient $\tilde{\boldsymbol{g}}$, we perturb $\boldsymbol{\mu}$ by adding Gaussian noise and recompute the gradient, as detailed in [21].

3) The Final Formula: The noise compensated step size is obtained using the following formula:

$$a = \eta \frac{\delta A^{\alpha}}{E\|\boldsymbol{M}_0(\boldsymbol{x}_j)\| + 2\sqrt{Var\|\boldsymbol{M}_0(\boldsymbol{x}_j)\|}}. \tag{19}$$

In summary, the gradient $\boldsymbol{g}$ is first calculated using (16), and then the magnitude $\boldsymbol{M}_0(\boldsymbol{x}_j)$ is computed at each voxel $\boldsymbol{x}_j$, finally $a_{\max}$ is obtained. In step 2, the noise compensation η is calculated through the perturbation process. Finally, a is obtained through (19).

E. Performance of Proposed Method

In this section, we compare the time complexity of the fast ASGD method with the ASGD method. Here we only give the final formula of the ASGD method, for more details see reference [21]. The ASGD method uses the following equation:

$$a_{\max} = \frac{\delta A^{\alpha}}{\sigma} \min_{\boldsymbol{x}_j \in \Omega_F^1} \left[Tr(\boldsymbol{J}_j \boldsymbol{C} \boldsymbol{J}_j') + 2\sqrt{2} \|\boldsymbol{J}_j \boldsymbol{C} \boldsymbol{J}_j'\|_F \right]^{-1/2}, \tag{20}$$

where σ is a scalar constant related to the distribution of the exact gradient $\boldsymbol{g}$ [21], $\boldsymbol{C} = (1/|\Omega_F^1|^2) \sum_j \boldsymbol{J}_j' \boldsymbol{J}_j$ is the covariance of the Jacobian, and $\|\cdot\|_F$ denotes the Frobenius norm.

From (13), the time complexity of FASGD is dominated by three terms: the Jacobian $\boldsymbol{J}(\boldsymbol{x}_j)$ with size $d \times P$, the gradient $\boldsymbol{g}$ of size P, and the number of voxels N_1 from which the expectation and variance of $\boldsymbol{M}_0$ are calculated. The matrix computation $\boldsymbol{M}_0(\boldsymbol{x}_j) = \boldsymbol{J}(\boldsymbol{x}_j)\boldsymbol{g}$ requires $d \times P$ multiplications and additions for each of the N_1 voxels $\boldsymbol{x}_j$, and therefore the time complexity of the proposed method is $\mathcal{O}(dN_1P)$. The dominant terms in (20) are the Jacobian (size $d \times P$) and its covariance matrix $\boldsymbol{C}$ (size $P \times P$). Calculating $\boldsymbol{J}_j \boldsymbol{C} \boldsymbol{J}_j'$ from right to left requires $d \times P^2$ multiplications and additions for $\boldsymbol{C}\boldsymbol{J}_j'$ and an additional $d^2 \times P$ operations for the multiplication with the left-most matrix $\boldsymbol{J}_j$. Taking into account the number of voxels N_1, the time complexity of the original ASGD method is therefore $\mathcal{O}(N_1 \times (d \times P^2 + d^2 \times P)) = \mathcal{O}(dN_1P^2)$, as $P \gg d$. This means that FASGD has a linear time complexity with respect to the dimension of $\boldsymbol{\mu}$, while ASGD is quadratic in P.

For the B-spline transformation model, the size of the nonzero part of the Jacobian is much smaller than the full Jacobian, i.e., only $d \times P_2$, where P_2 is determined by the B-spline order used in this model. For a cubic B-spline transformation model, each voxel is influenced by 4^d control points, so $P_2 = 4^2 = 16$ in 2D and $P_2 = 4^3 = 64$ in 3D. For the fast ASGD method the time complexity reduces to $\mathcal{O}(dN_1P_2)$ for the cubic B-spline model. However, as the total number of operations for the calculation of $\boldsymbol{J}_j \boldsymbol{C} \boldsymbol{J}_j'$ is still $d \times P_2 \times P$, the time complexity of ASGD is $\mathcal{O}(dN_1P_2P)$. Since $P \gg N_1 \geq P_2 > d$, the dominant term of FASGD becomes the number of samples N_1, while for ASGD it is still a potentially very large number P.

III. Data Sets

In this section we describe the data sets that were used to evaluate the proposed method. Data sets were chosen to represent a broad category of use cases, i.e., mono-modal and multi-modal, intra-patient as well as inter-patient, from different anatomical sites, and having rigid as well as nonrigid underlying deformations. The overview of all data sets is presented in Table I.

A. RIRE Brain Data – Multi-Modality Rigid Registration

The Retrospective Image Registration Evaluation (RIRE) project provides multi-modality brain scans with a ground truth for rigid registration evaluation [39]. These brain scans were obtained from 9 patients, where we selected CT scans and MR T1 scans. Fiducial markers were implanted in each patient, and served as a ground truth. These markers were manually erased from the images and replaced with a simulated background pattern.

In our experiments, we registered the T1 MR image (moving image) to the CT image (fixed image) using rigid registration. At the website of RIRE, eight corner points of both CT and MR T1 images are provided to evaluate the registration accuracy.

B. Spread Lung Data – Intra-Subject Nonrigid Registration

During the SPREAD study [40], 3D lung CT images of 19 patients were scanned without contrast media using a Toshiba

Aquilion 4 scanner with scan parameters: 135 kVp; 20 mAs per rotation; rotation time 0.5 s; collimation: 4 × 5 mm. Images were reconstructed with a standardized protocol optimized for lung densitometry, including a soft FC12 kernel, using a slice thickness of 5 mm and an increment of 2.5 mm, with an inplane resolution of around 0.7 × 0.7 mm. The patient group, aging from 49 to 78 with 36%-87% predicted FEV_1 had moderate to severe COPD at GOLD stage II and III, without $\alpha 1$ antitrypsin deficiency.

One hundred anatomical corresponding points from each lung CT image were semi-automatically extracted as a ground truth using Murphy's method [41]. The algorithm automatically finds 100 evenly distributed points in the baseline, only at characteristic locations. Subsequently, corresponding points in the follow-up scan are predicted by the algorithm and shown in a graphical user interface for inspection and possible correction. More details can be found in [42].

C. Hammers Brain Data – Inter-Subject Nonrigid Registration

We use the brain data set developed by Hammers *et al.* [43], which contains MR images of 30 healthy adult subjects. The median age of all subjects was 31 years (range 20 ~ 54), and 25 of the 30 subjects were strongly right handed as determined by routine pre-scanning screening. MRI scans were obtained on a 1.5 Tesla GE Sigma Echospeed scanner. A coronal T1 weighted 3D volume was acquired using an inversion recovery prepared fast spoiled gradient recall sequence (GE), TE/TR/NEX 4.2 msec (fat and water in phase)/15.5 msec/1, time of inversion (TI) 450 msec, flip angle 20°, to obtain 124 slices of 1.5 mm thickness with a field of view of 18 × 24 cm with a 192 × 256 matrix [44]. This covers the whole brain with voxel sizes of 0.94 × 0.94 × 1.5 mm^3. Images were resliced to create isotropic voxels of 0.94 × 0.94 × 0.94 mm^3, using windowed sinc interpolation.

Each image is manually segmented into 83 regions of interest, which serve as a ground truth. All structures were delineated by one investigator on each MRI in turn before the next structure was commenced, then a separate neuroanatomically trained operator evaluated each structure to ensure that consensus was reached for the difficult cases. In our experiment, we performed inter-subject registration between all patients. Each MR image was treated as a fixed image as well as a moving image, so the total number of registrations for 30 patients was 870 for each particular parameter setting.

D. Ultrasound Data – 4D Nonrigid Registration

We used the 4D abdominal ultrasound dataset provided by Vijayan *et al.* [45], which contains 9 scans from three healthy volunteers at three different positions and angles. Each scan was taken over several breathing cycles (12 seconds per cycle). These scans were performed on a GE Healthcare vivid E9 scanner by a skilled physician using an active matrix 4D volume phased array probe.

The ground truth is 22 well-defined anatomical landmarks, first indicated in the first time frame by the physician who acquired the data, and then manually annotated in all 96 time frames by engineers using VV software [46].

IV. Experiment Setup

In this section, the general experimental setup and the evaluation measurements are presented and more details about the experimental environment are given.

A. Experimental Setup

The experiments focus on the properties of the fast ASGD method in terms of registration accuracy, registration runtime and convergence of the algorithm. We will compare the proposed method with two variants of the original ASGD method. While for FASGD $f_{\min}$ and ω are fixed, the ASGD method automatically estimates them. For a fair comparison, a variant of the ASGD method is included in the comparison, that sets these parameters to the same value as FASGD: $f_{\min} = -0.8$ and $\omega = 10^{-8}$. In summary, three methods are compared in all the experiments: the original ASGD method that automatically estimates all parameters (ASGD), the ASGD method with default settings only estimating a (ASGD$'$) and the fast ASGD method (FASGD). The fast ASGD method has been implemented using the C++ language in the open source image registration toolbox `elastix` [37], where the ASGD method is already integrated.

To thoroughly evaluate FASGD, a variety of imaging problems including different modalities and different similarity measures are considered in the experiments. Specifically, the experiments were performed using four different datasets, rigid and nonrigid transformation models, inter/intra subjects, four different dissimilarity measures and three imaging modalities. The experiments are grouped by the experimental aim: registration accuracy in Section V-A, registration time in Section V-B and algorithm convergence in Section V-C. The RIRE brain data is used for the evaluation of rigid registration. The SPREAD lung CT data is especially used to verify the performance of FASGD on four different dissimilarity measures, including the mean squared intensity difference (MSD) [2], normalized correlation (NC) [2], mutual information (MI) [12] and normalized mutual information (NMI) [47]. The Hammers brain data is intended to verify inter-subject registration performance. The ultrasound data is specific for 4-dimensional medical image registration, which is more complex. An overview of the experimental settings is given in Table I.

For the evaluation of the registration accuracy, the experiments on the RIRE brain data, the SPREAD lung CT data and the ultrasound abdominal data, were performed on a local workstation with 24 GB memory, Linux Ubuntu 12.04.2 LTS 64 bit operation system and an Intel Xeon E5620 CPU with 8 cores running at 2.4 GHz. To see the influence of the parameters A and δ on the registration accuracy, we perform an extremely large scale experiment on the Hammers brain data using the Life Science Grid (`lsgrid`) [48], which is a High Performance Computing (HPC) facility. We tested all combinations of the following settings: $A \in \{1.25, 2.5, \ldots, 160, 320\}$, $\delta \in \{0.03125, 0.0625, \ldots, 128, 256\}$ (in mm) and $k \in \{250, 2000\}$. This amounts to 252 combinations of registration settings and a total of 657,720 registrations, see Table I. Each registration requires about 15 minutes of computation time, which totals about 164,000 core hours of computation, i.e ~ 19 years, making the use of an HPC resource essential. With the `lsgrid` the run time of the Hammers experiment is

TABLE I
OVERVIEW OF DATA SETS AND EXPERIMENTS.

	RIRE	SPREAD	Hammers	Abdomen
Anatomy	Brain	Lung	Brain	Abdomen
Modality	CT and 1.5T MR T1	CT	MR	Ultrasound
Dimensions	CT: $512 \times 512 \times 50$ MR: $256 \times 256 \times 50$	3D: $450 \times 300 \times 130$	3D: $180 \times 200 \times 170$	4D: $227 \times 229 \times 227 \times 96$
Voxel size (mm)	CT: $0.45 \times 0.45 \times 3$ MR: $0.85 \times 0.85 \times 3$	$\sim 0.7 \times 0.7 \times 2.5$	$0.94 \times 0.94 \times 0.94$	$0.7 \times 0.7 \times 0.7 \times 1$
Number of patients	9	21	30	3 volunteers × 3 positions
Registration	Multi-modality Intra subject	Single modality Intra subject	Single modality Inter subject	Single modality Intra subject
Similarity measure	MI	MSD, NC, MI, NMI	MI	MI
Transformation	Rigid	Affine + B-spline	Similarity + B-spline	B-spline
B-spline control point grid spacing (mm)	-	$10 \times 10 \times 10$	$5 \times 5 \times 5$	$15 \times 15 \times 15 \times 1$
Number of parameters (last resolution)	6	$\sim 90k$	$\sim 150k$	$\sim 870k$
Ground truth	8 corner points	100 corresponding points	83 labelled regions	22 landmarks
Evaluation measure	Euclidean distance	Euclidean distance	Dice overlap	Euclidean distance
Number of registrations per setting	9×3	19×3	$30 \times 29 \times 3$	9×3
Settings	1	1	252	1
Total number of registrations	27	228	657,720	27

reduced to 2–3 days. More details about the `lsgrid` are given in the Appendix.

For a fair comparison, all timing experiments were carried out on the local workstation. Timings are reported for all the registrations, except for the Hammers data set, where we only report timings from a subset. From (19), we know that the runtime is independent of the parameters A and δ. Therefore, for the Hammers data, we used $A = 20$ and δ equal to the voxel size. We randomly selected 100 out of the 870 registrations, as a sufficiently accurate approximation.

The convergence of the algorithms is evaluated in terms of the step size, the Euclidean distance error and the cost function value, as a function of the iteration number.

All experiments were done using the following routine: (1) Perform a linear registration between fixed and moving image to get a coarse transformation $\boldsymbol{T}_0$, using a rigid transformation for the RIRE brain data, an affine transformation for the SPREAD lung CT data, a similarity transformation rigid plus isotropic scaling for the Hammers brain data, and no initial transformation for the 4D ultrasound data; (2) Perform a non-linear cubic B-spline based registration [49] for all datasets except the RIRE data to get the transformation $\boldsymbol{T}_1$. For the ultrasound data, the B-spline transformation model proposed by Metz *et al.* [50] is used, which registers all 3D image sequences in a group-wise strategy to find the optimal transformation that is both spatially and temporally smooth. A more detailed explanation of the registration methodology is in [45]; (3) Transform the landmarks or moving image segmentations using $\boldsymbol{T}_1 \circ \boldsymbol{T}_0$; (4) Evaluate the results using the evaluation measures defined in Section IV-B.

For each experiment, a three level multi-resolution strategy was used. The Gaussian smoothing filter had a standard deviation of 2, 1 and 0.5 mm for each resolution. For the B-spline transformation model, the grid size of the B-spline control point mesh is halved in each resolution to increase the transformation accuracy [49]. We used $K = 500$ iterations and 5000 samples, except for the ultrasound experiment where we used 2000 iterations and 2000 samples according to Vijayan [45]. We set $A = 20$ and δ equal to the voxel size (the mean length of the voxel edges).

B. Evaluation Measures

Two evaluation measures were used to verify the registration accuracy: the Euclidean distance and the mean overlap. The Euclidean distance measure is given by:

$$\mathrm{ED} = \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{T}(\boldsymbol{p}_F^i) - \boldsymbol{p}_M^i\|, \tag{21}$$

in which $\boldsymbol{p}_F^i$ and $\boldsymbol{p}_M^i$ are coordinates from the fixed and moving image, respectively. For the RIRE brain data, 8 corner points and for the SPREAD data 100 corresponding points are used to evaluate the performance. For the 4D ultrasound image, we adopt the following measure from [45]:

$$\mathrm{ED} = \left(\frac{1}{\tau - 1} \sum_t \|\boldsymbol{p}_t - \boldsymbol{T}_t(\boldsymbol{q})\|^2 \right)^{1/2}, \tag{22}$$

in which $\boldsymbol{p}_t = 1/J \sum_j \boldsymbol{p}_{tj}$ and $\boldsymbol{p}_{tj}$ is a landmark at time t placed by observer j, $\boldsymbol{q} = 1/\tau \sum_t \boldsymbol{S}_t(\boldsymbol{p}_t)$ is the mean of landmarks after inverse transformation.

The mean overlap of two segmentations from the images is calculated by the Dice Similarity Coefficient (DSC) [5]:

$$\mathrm{DSC} = \frac{1}{R} \sum_r \frac{2|\boldsymbol{M}_r \cap \boldsymbol{F}_r|}{|\boldsymbol{M}_r| + |\boldsymbol{F}_r|}, \tag{23}$$

in which r is a labelled region and $R = 83$ the total number of regions for the Hammers data.

To assess the registration accuracy, a Wilcoxon signed rank test ($p = 0.05$) for the registration results was performed. For the SPREAD data, we first obtained the mean distance error of 100 points for each patient and then performed the Wilcoxon signed rank test to these mean errors.

Registration smoothness is assessed for the SPREAD experiment by measuring the determinant of the spatial Jacobian of the transformation, $J = |\partial \boldsymbol{T} / \partial \boldsymbol{x}|$ [51]. Because the fluctuation

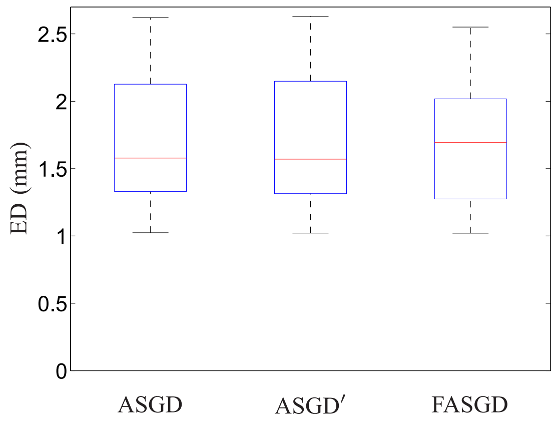


Fig. 1. Euclidean distance error in mm for the RIRE brain data performed using MI.

TABLE II
THE MEDIAN EUCLIDEAN DISTANCE ERROR (MM) FOR THE SPREAD LUNG CT DATA. THE SYMBOLS † AND ‡ INDICATE A STATISTICALLY SIGNIFICANT DIFFERENCE WITH ASGD AND ASGD′, RESPECTIVELY. × DENOTES NO SIGNIFICANT DIFFERENCE.

	Initial	ASGD	ASGD′	FASGD
MSD	3.62	1.09	1.10 ×	1.12 † ‡
NC	3.56	1.50	1.51 †	1.55 × ×
MI	3.17	1.65	1.65 †	1.66 † ‡
NMI	3.17	1.66	1.65 ×	1.68 † ‡

of J should be relatively small for smooth transformations, we use the standard deviation of J to represent smoothness.

The computation time is determined by the number of parameters and the number of voxels sampled from the fixed image. For a small number of parameters the estimation time can be ignored, and therefore we only provide the comparison for the B-spline transformation. Both the parameter estimation time and pure registration time were measured, for each resolution.

V. RESULTS

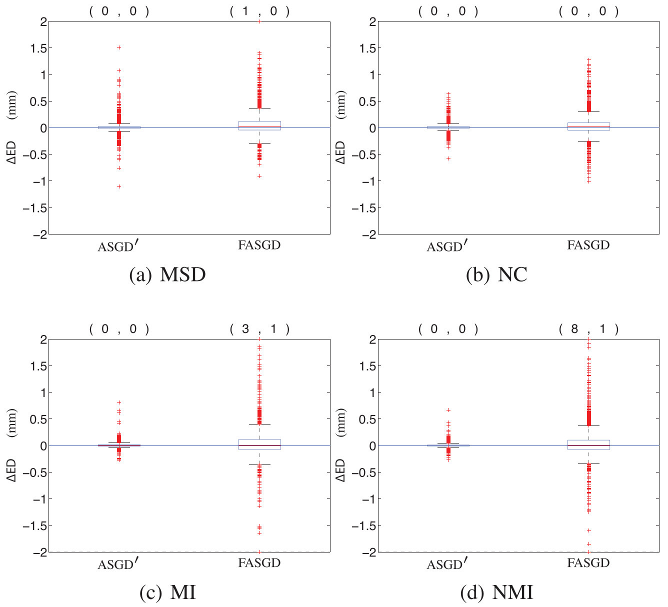
A. Accuracy Results

In this section, we compare the registration accuracy between ASGD, ASGD′ and FASGD.

1) RIRE Brain Data: The results shown in Fig. 1 present the Euclidean distance error of the eight corner points from the brain images. The median Euclidean distance before registration is 21.7 mm. The result of the FASGD method is very similar to the ASGD method: median accuracy is 1.6, 1.6 and 1.7 mm for ASGD, ASGD′ and FASGD, respectively. The p value of the Wilcoxon signed rank test of FASGD compared with ASGD and ASGD′ is 0.36 and 0.30, respectively, indicating no statistical difference.

2) Spread Lung CT Data: Table II shows the median of the mean Euclidean distance error of the 100 corresponding points of 19 patients for four different similarity measures. Compared with ASGD, FASGD has a significant difference for MSD, MI and NMI, but the median error difference is smaller than 0.03 mm.

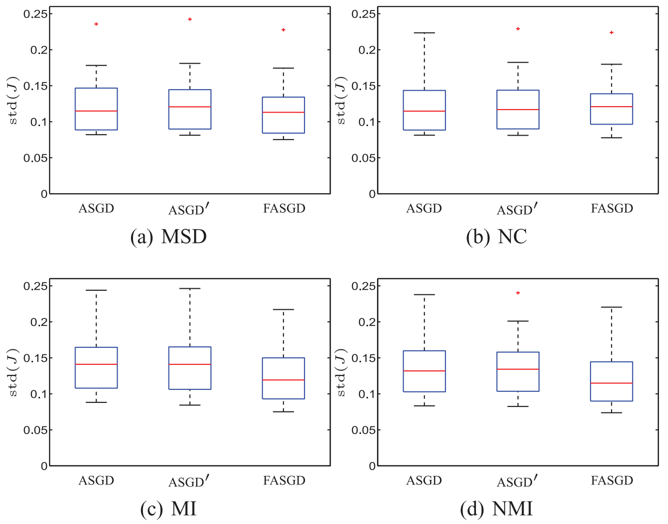
To compare FASGD and ASGD′ with ASGD we define the Euclidean landmark error difference as $\Delta\mathrm{ED}_i = \mathrm{ED}_i^{\mathrm{FASGD}} - \mathrm{ED}_i^{\mathrm{ASGD}}$, for each landmark i, and similarly for ASGD′. This difference is shown as a box plot in Fig. 2. Negative numbers mean that FASGD is better than ASGD, and vice versa. It can be seen that both ASGD′ and FASGD provide results similar to ASGD, for all tested cost functions. The spread of the ΔED box plot for ASGD′ is smaller than that of FASGD, as this method is almost identical to ASGD.



(a) MSD (b) NC

(c) MI (d) NMI

Fig. 2. The difference of Euclidean distance error in mm compared to ASGD for the SPREAD lung CT data. The two numbers on the top of each box denote the number of the landmark errors larger (left) and smaller (right) than 2 and -2 mm, respectively. All those landmarks, except one for NMI, belong to the same patient.



(a) MSD (b) NC

(c) MI (d) NMI

Fig. 3. Box plots of the standard deviation of the Jacobian determinant J for the four similarity measures.

Smoothness of the resulting transformations is given in Fig. 3 for all similarity measures. FASGD generates somewhat smoother transformations over ASGD and ASGD′ for the MSD, MI and NMI measures.

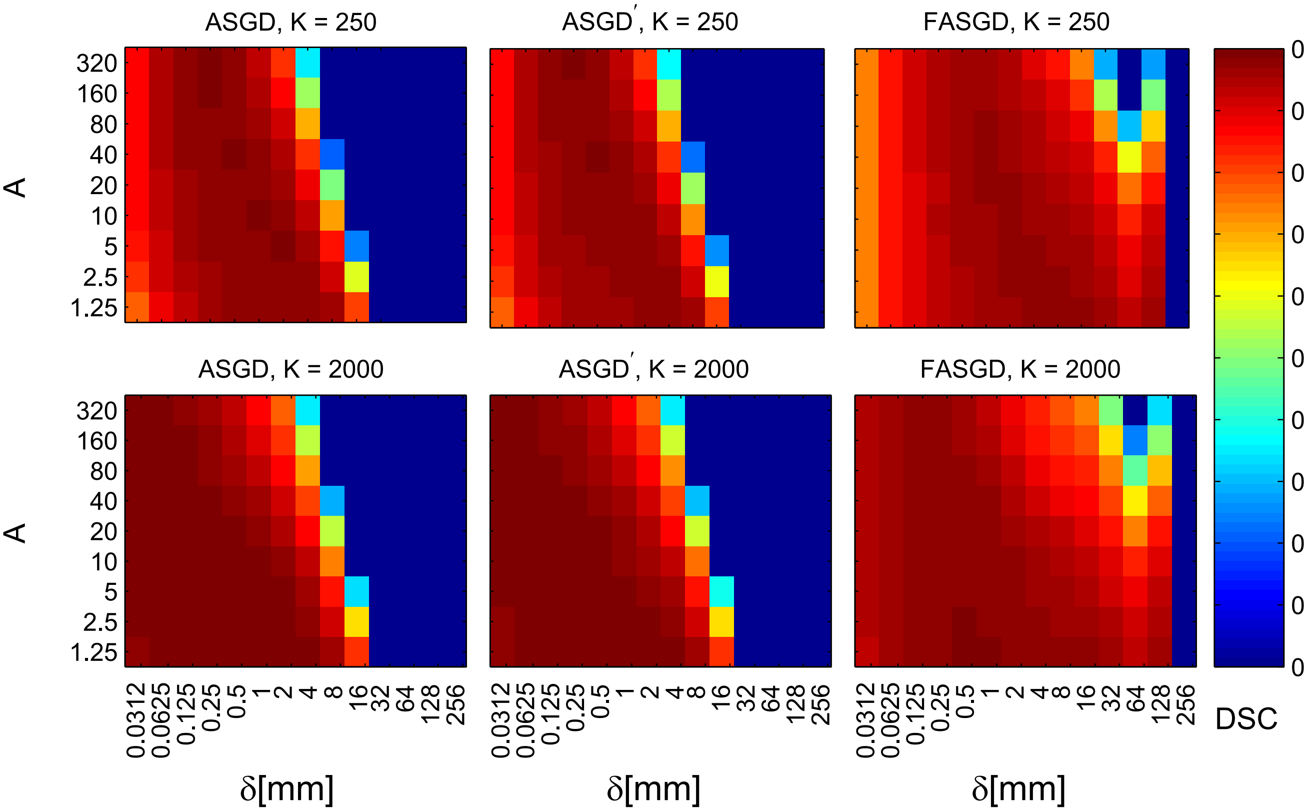


Fig. 4. Median dice overlap after registration of the Hammers brain data, as a function of A and δ. A high DSC indicates better registration accuracy. Note that in this large scale experiment, each square represents 870 registrations, requiring about 870 × 15 minutes of computation, i.e., almost 200 core hours.

3) Hammers Brain Data: In this experiment, FASGD is compared with ASGD and ASGD$'$ in a large scale intersubject experiments on brain MR data, for a range of values of A, δ and the number of iterations K.

Fig. 4 shows the overlap results of the 83 brain regions. Each square represents the median DSC result of 870 brain image registration pairs for a certain parameter combination of A, δ and K. These results show that the original ASGD method has a slightly higher DSC than FASGD with the same parameter setting, but the median DSC difference is smaller than 0.01. Note that the dark black color indicates DSC values between 0 and 0.5, i.e., anything between registration failure and low performance. The ASGD and ASGD$'$ methods fail for $\delta \geq 32$ mm, while FASGD fails for $\delta \geq 256$ mm.

4) Ultrasound Abdomen Data: The results shown in Fig. 5 present the Euclidean distance of 22 landmarks from ultrasound images after nonrigid registration. The median Euclidean distance before registration is 3.6 mm. The result of FASGD is very similar to the original method. The p value of the Wilcoxon signed rank test of FASGD compared with ASGD and ASGD$'$ is 0.485 and 0.465, respectively, indicating no statistical difference.

B. Runtime Results

In this section the runtime of the three methods, ASGD, ASGD$'$ and FASGD is compared.

1) Spread Lung CT Data: The runtime on SPREAD lung CT data is shown in Fig. 6, in which the time used in the estimations of the original method takes a large part of the total runtime per resolution, while FASGD consumes only a small fraction of the total runtime. From resolution 1 (R1) to resolution 3 (R3), the number of transformation parameters P increases from 4×10^3 to 9×10^4. For both ASGD and ASGD$'$ the estimation time increases from 3 seconds in R1 to 40 seconds in R3. However, FASGD maintains a constant estimation time of no more than 1 second.

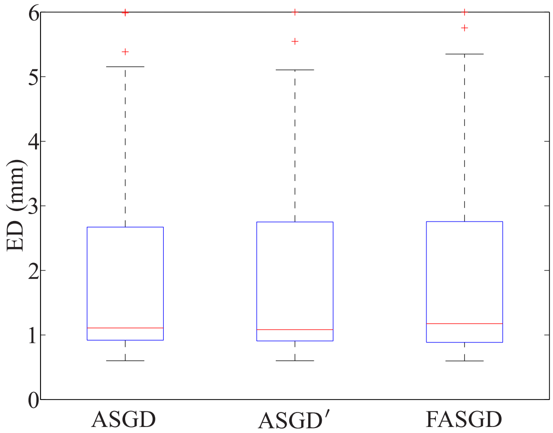
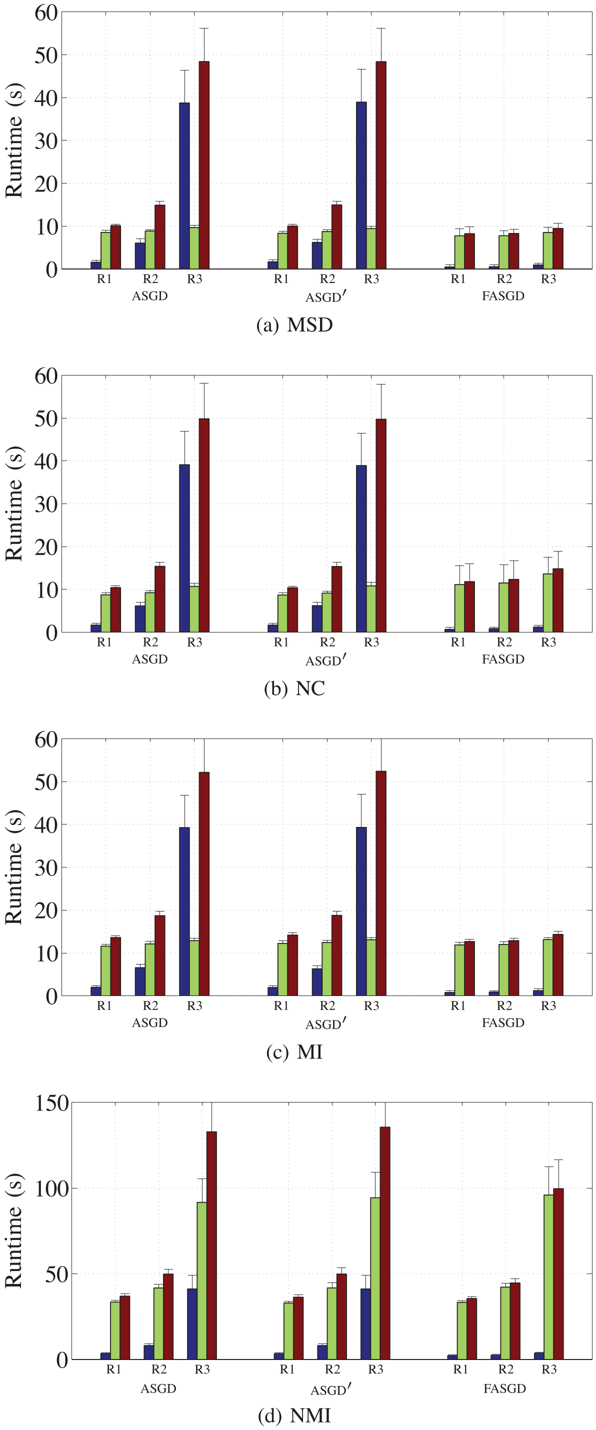


Fig. 5. Euclidean distance in mm of the registration results for Ultrasound data performed using MI.

2) Hammers Brain Data: The runtime result of the Hammers brain data is shown in Fig. 7. For this dataset, $P \approx 1.5 \times 10^5$ in R3, i.e., larger than for the SPREAD data, resulting in larger estimation times. For ASGD and ASGD$'$ the estimation time in



(a) MSD

(b) NC

(c) MI

(d) NMI

Fig. 6. Runtime of SPREAD lung CT data in seconds. The black, green and red bar indicate estimation time, pure registration time and total time elapsed in each resolution, respectively. R1, R2, R3 indicate a three level multi-resolution strategy from low resolution to high resolution.

the third resolution is almost 95 seconds, while for FASGD it is almost 2 orders of magnitude smaller (≤ 1 s).

3) 4D Ultrasound Data: The grid spacing of B-spline control points used in the 4D ultrasound data experiment is $15 \times 15 \times 15 \times 1$ and the image size is $227 \times 229 \times 227 \times 96$, so the total number of B-spline parameters for the third resolution R3 is around 8.7×10^5. From the timing results in Fig. 8, the original method takes almost 1400 seconds, i.e., around 23 minutes, while FASGD only takes 40 seconds.

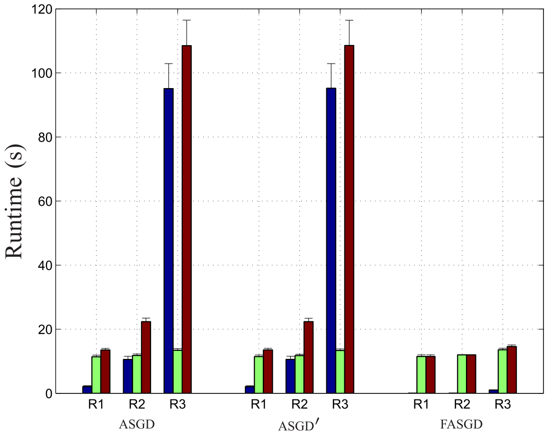


Fig. 7. Runtime of Hammers brain data experiment in seconds. The black, green and red bar indicate estimation time, pure registration time and total time elapsed in each resolution, respectively. R1, R2, R3 indicate a three level multi-resolution strategy from low resolution to high resolution.

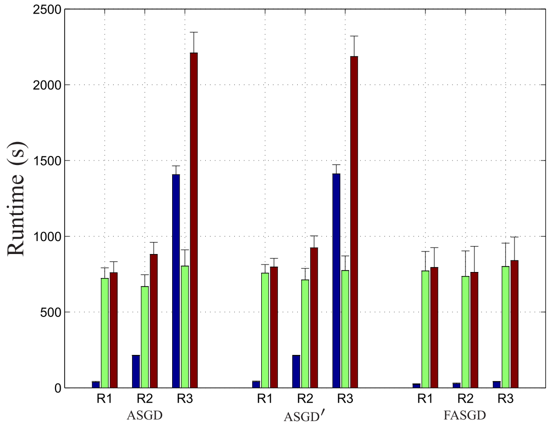


Fig. 8. Runtime of Ultrasound data experiment in seconds. The black, green and red bar indicate estimation time, pure registration time and total time elapsed in each resolution, respectively. R1, R2, R3 indicate a three level multi-resolution strategy from low resolution to high resolution.

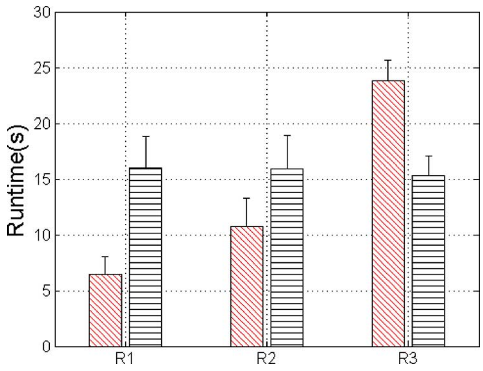
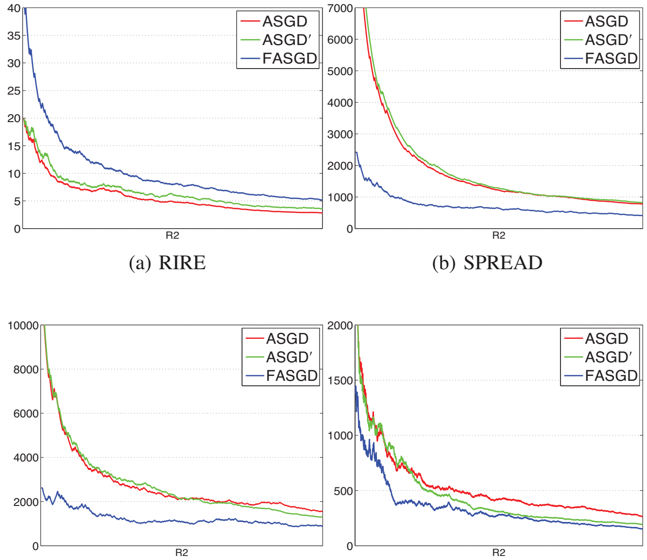


Fig. 9. Runtime in seconds of FASGD for ultrasound experiment. The left bar indicate estimation time of $a_{\max}$ and the right bar is the estimation time of η. R1, R2, R3 indicate a three level multi-resolution strategy from low resolution to high resolution.

Fig. 9 presents the runtime of estimating $a_{\max}$ and η for the ultrasound data. The estimation of η takes a constant time during each resolution, so for small P the estimation of η dominates the total estimation time.



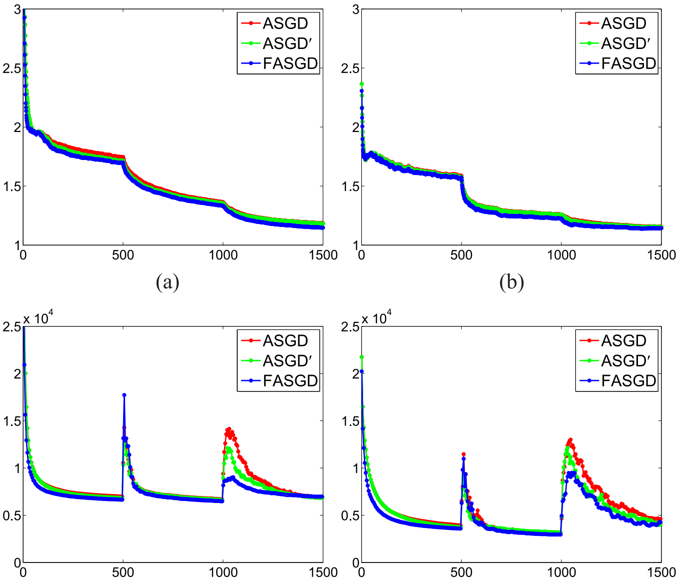
(a) RIRE

(b) SPREAD

(c) Hammers

(d) Ultrasound

Fig. 10. An example of the step size decay using 500 iterations except Ultrasound image data (2000 iterations) in last resolution from four experiments. The red line is the original ASGD, the black line is ASGD′ and the green line is FASGD. (a) RIRE. (b) SPREAD. (c) Hammers. (d) Ultrasound.



(a)

(b)

(c)

(d)

Fig. 11. Convergence plots for four different patients. Top row shows the Euclidean distance error (mm) as a function of the iteration number. Bottom row shows the cost function value (MSD). Each plot shows three resolutions. (a) ED, patient 1. (b) ED, patient 2. (c) MSD, patient 3. (d) MSD, patient 4.

C. Convergence

From each of the four experiments, we randomly selected one patient and analyzed the step size sequence $\{\gamma_k\}$. The results are presented in Fig. 10 and show that FASGD takes a larger step size than ASGD and ASGD′ for rigid registration and a smaller step size for nonrigid registration, when using the same δ. In addition, the original ASGD and ASGD′ take a very similar step size in all experiments even when ASGD′ uses the default settings for $f_{\min}$ and ω.

Convergence results of the three methods are presented in Fig. 11 for several patients. Fig. 11(a) and (b) present the Euclidean distance (mm) at each iteration for three resolutions with respect to the iteration number. The cost function values are shown in Fig. 11(c) and (d). The three methods behave similarly.

VI. Discussion

All experiments in this paper show that the fast ASGD method works well both in rigid and nonrigid image registration, showing that the method can deal with differently parameterized transformations. The method was thoroughly evaluated on a variety of imaging problems, including different modalities such as CT, MRI and ultrasound, intra and inter subject registration, and different anatomical sites such as the brain, lung and abdomen. Various image registration settings were tested, including four popular similarity measures. A very large scale experiment investigated the sensitivity of the methods to the parameters A and δ.

All experiments show that FASGD has similar accuracy as the ASGD method. For the rigid registration on the RIRE data and the nonrigid 4D ultrasound experiment there was no significant statistical difference. For the nonrigid SPREAD lung CT experiment and the Hammers brain data we observed statistically significant differences, however, these differences were very small: on average less than 0.03 mm on the SPREAD data (less than 5% of the voxel size), and less than 0.01 Dice overlap on brain data. We conclude that FASGD obtains a very similar registration accuracy as the original ASGD method.

All results indicate that there is little difference between ASGD and ASGD′. Especially from Fig. 10 it can be observed that both methods take very similar step size during the optimization, as well as similar cost function value and Euclidean distance error (Fig. 11). This suggests that the default values of the parameters $f_{\min}$ and ω are sufficiently accurate, and that indeed the parameter a is the most important parameter to estimate.

From Fig. 10 it can be observed that FASGD typically estimates smaller step sizes than ASGD, for identical δ. This was also observed for the other patients. Fig. 4 confirms this observation, as the accuracy plot for FASGD is somewhat shifted to the right compared to the other two methods. This suggests that more similar step sizes may be obtained when choosing δ about twice as large as for ASGD, i.e., to increase the default from one voxel size to two.

The accuracy results for the Hammers experiment shown in Fig. 4 present an apparent accuracy increase when $\delta = 128$ for FASGD. Remember that δ represents the maximum allowed voxel displacement per iteration in mm, and that for the medical data used in this paper larger δ are unrealistic. Note that for ASGD the registrations start failing when $\delta \geq 32$, and for FASGD when $\delta > 128$. The temporary increase in accuracy at $\delta = 128$ for FASGD is due to an undesired decrease in $\eta \times \delta$. Note that ASGD uses the exact same term, see (20), but this does not result in increased accuracy, since ASGD is already failing for $\delta = 128$.

The time performance of the proposed method shown in Section V-B implies that FASGD has a large reduction in time consumption of the step size estimation. For the SPREAD experiment the estimation time in the last resolution is reduced from 40 seconds to 1 second. This improvement is crucial for near real-time registration in high dimensional image registration.

From Fig. 9 it is observed that a new bottle neck in the step size estimation is the estimation of the noise compensation parameter η. This is because in this work the calculation of the gradient $\boldsymbol{g}$ is performed with a relatively high number of voxels from the fixed image. Future work will include the investigation of accelerated methods to estimate η and so further reduce the step size estimation time, especially for 4D registration problems. A direct acceleration possibility is the use of parallelization, for example by a GPU implementation, as the gradient computation consists of an independent loop over the voxels.

The FASGD method provides a solution for step size selection for gradient descent optimizers. For Newton-like optimizers this is typically solved by a line search strategy. Note that such a strategy can not readily be adopted for stochastic optimization due to the stochastic approximation of the cost function [52]. Strengths of quasi-Newton optimizers are their adaptability to problems where the parameters are scaled with respect to each other, and the availability of stopping conditions. For FASGD as well as other stochastic gradient descent optimization routines typically the number of iterations is used to terminate the optimization. More sophisticated stopping conditions from deterministic gradient descent methods cannot be readily adopted. For example, due to the estimation noise, stopping conditions based on cost function values or cost function gradients cannot be trusted. The alternative to compute exact objective values every (few) iteration(s), is also not attractive due to the required computation time. In the `elastix` implementation a stochastic gradient computation is in the order of 50 ms, while exact metric value computation is at least in the order of seconds. A feasible possibility would be to create a stopping condition based on a moving average of the noisy objective values or gradients.

The use of the `lsgrid` for the Hammers data experiment was essential, and reduced computation time from 19 years to about 2–3 days. It however did require a one-time investment of time to develop the software supporting the registration jobs on the grid. Typical issues we encountered was attempting to store the results from hundreds of simultaneous executions, which proved incompatible with maximum transaction rate supported by the `lsgrid` Storage Resource Management services. We were able to solve this by pooling multiple results into a single storage operation. The infrastructure we built therefore screens the software under execution from the complexities that are encountered when running on the `lsgrid`. At the same time it is generic enough to provide a configurable set of execution environments to support other experiments not just the `elastix` workflow used in this work, and can therefore be re-used.

VII. Conclusion

In this paper, a new automatic method (FASGD) for estimating the optimization step size parameter a, needed for gradient descent optimization methods, has been presented for image registration. The parameter a is automatically estimated from the magnitude of voxel displacements, randomly sampled from the fixed image. A relation between the step size and the expectation and variance of the observed voxels displacement is derived. The proposed method has a free parameter δ, defining the maximally allowed incremental displacement between iterations. Unlike a, it can be interpreted in terms of the voxel size (mm). In addition, it is mostly independent of the application domain, i.e., setting it equal to the voxel size provided good results for all applications evaluated in this paper. Compared to the original ASGD method, the time complexity of the FASGD method is reduced from quadratic to linear with respect to the dimension of the transformation parameters P. For the B-spline transformation, due to its compact support, the time complexity is further reduced, making the proposed method independent of P. The FASGD method is publicly available via the open source image registration toolbox `elastix` [37].

The FASGD method was evaluated on a large number of registration scenario's and shows a similar accuracy as the original ASGD method. It however improves the time complexity of the step size estimation from 40 seconds to no more than 1 second, when the number of parameters is $\sim 10^5$: almost 40 times faster. Depending on the registration settings, the total registration time is reduced by a factor of 2.5–7 × for the experiments in this paper.

Appendix

The `lsgrid` infrastructure comprises distributed computing and storage resources along with a central grid facility. In total there is potential for approximately 10000 job slots. Job scheduling is performed using gLite grid middleware [53] via the gLite Workload Management System (WMS) [54], which was developed for the European Grid Infrastructure [55].

While it is possible to use this directly to schedule registration pipeline jobs, in practice these relatively short jobs are a poor fit to the standard queue lengths in `lsgrid`. In addition, unforseen delays in the push scheduling mechanism result in a considerable overhead [56]. These issues can be addressed by layering a pull scheduling system based on pilot jobs onto the grid software infrastructure. Matching jobs to Workload Nodes occurs once at pilot job startup after which job tokens are pulled into the pilot job environment. The concept of Pilot Jobs was first pioneered in the EGI grid within DIRAC [57], but we employed a light weight pilot job system developed by SURFsara called PiCaS [58], [59].

The pilot job architecture shown in Fig. 12 was used to execute the Hammers pipeline. PiCaS was extended with a wrapper job to perform standard elements of the pipeline such as environment setup and data retrieval. The wrapper job and the Hammers pipeline are coded using Python [60]. The job tokens contain the registration parameters to be used and the storage locations for the fixed and moving images. Ganga [61] is used to schedule and monitor pilot jobs which pull and execute the job-tokens from the PiCaS database. The overall progress of the execution can be checked by monitoring the status of the job tokens using the web browser to access job-token views defined in database.

Execution of the Hammers pipeline using PiCaS on the `lsgrid` follows these steps:

1) Initialize the Hammers jobs tokens. (a) Create the job tokens for each Hammers pipeline run. Job tokens contain

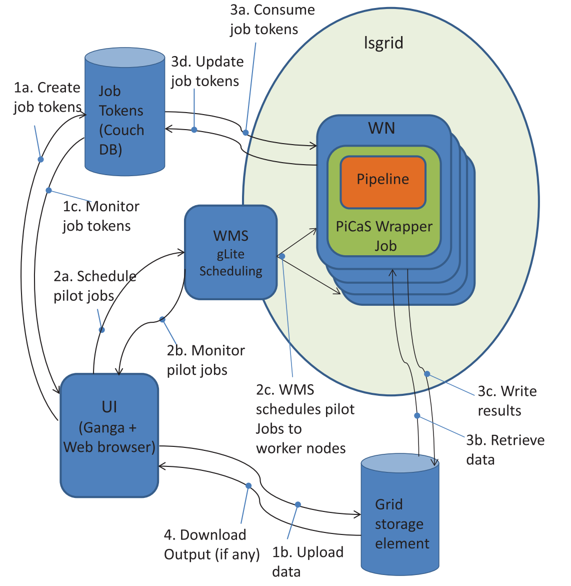


Fig. 12. Running the Hammers pipeline in the pilot job architecture used on the lsgrid. Arrows represent the flow of information.

job parameters and the grid location of the input data. (b) Upload the input data needed to specific locations in grid storage. (c) Monitor execution progress by checking job token consumption in a browser.

2) Schedule the pilot jobs to commence grid execution. (a) Schedule pilot jobs with the necessary job requirements using gLite WMS from inside Ganga. Additional information is passed to the pilot job concerning the runtime environment needed. (b) Monitor the progress of the pilot jobs using Ganga job monitoring. (c) gLite WMS identifies clusters matching the job requirements and schedules pilot jobs. Once the pilot is started the PiCaS Wrapper Job sets up the runtime environment on the worker node.
3) Job tokens are consumed and executed by the running pilot jobs. (a) Retrieve a job token from the PiCaS job tokens database and mark it as locked. (b) The necessary data identified in the job token for each Hammers job is downloaded by the PiCaS wrapper from grid storage and the Hammers pipeline is executed. (c) Any results are uploaded to the grid storage location as specified in the job token. (d) The job token is updated with the result: success or failure. In failure cases log-files are appended to assist in debugging.
4) Job results can be immediately downloaded while the run is in progress.

All tools that were created are reusable for other large scale image processing with the `lsgrid`.

Acknowledgment

The RIRE project is acknowledged for providing a platform for rigid registration evaluation. The authors are grateful to Dr. A. Hammers *et al.* for the adult maximum probability brain atlas. Dr. M.E. Bakker and J. Stolk are acknowledged for providing a ground truth for the SPREAD study data used in this paper. The 4D ultrasound data was made available by SINTEF Dept. Medical Technology and the Norwegian University of Science and Technology in the context of the IIIOS project (Marie Curie ITN no 238802).

References

[1] J. Maintz and M. A. Viergever, "A survey of medical image registration," *Med. Image Anal.*, vol. 2, no. 1, pp. 1–36, 1998.
[2] B. Zitova and J. Flusser, "Image registration methods: A survey," *Image Vis. Comput.*, vol. 21, no. 11, pp. 977–1000, 2003.
[3] J. P. Pluim, J. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: A survey," *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 986–1004, Aug. 2003.
[4] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *IEEE Trans. Med. Imag.*, vol. 32, no. 7, pp. 1153–1190, Jul. 2013.
[5] A. Klein *et al.*, "Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration," *Neuroimage*, vol. 46, no. 3, pp. 786–802, 2009.
[6] R. Shams, P. Sadeghi, R. A. Kennedy, and R. I. Hartley, "A survey of medical image registration on multicore and the GPU," *IEEE Signal Process. Mag.*, vol. 27, no. 2, pp. 50–60, Mar. 2010.
[7] D. P. Shamonin *et al.*, "Fast parallel image registration on CPU and GPU for diagnostic classification of Alzheimer's disease," *Front. Neuroinformat.*, vol. 7, 2013.
[8] F. Maes, D. Vandermeulen, and P. Suetens, "Comparative evaluation of multiresolution optimization strategies for multimodality image registration by maximization of mutual information," *Med. Image Anal.*, vol. 3, no. 4, pp. 373–386, 1999.
[9] J. Kybic and M. Unser, "Fast parametric elastic image registration," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1427–1442, Nov. 2003.
[10] S. Klein, M. Staring, and J. P. Pluim, "Evaluation of optimization methods for nonrigid medical image registration using mutual information and B-splines," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2879–2890, Dec. 2007.
[11] R. C. Hardie, K. J. Barnard, and E. E. Armstrong, "Joint map registration and high-resolution image estimation using a sequence of undersampled images," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1621–1633, Dec. 1997.
[12] P. Thevenaz and M. Unser, "Optimization of mutual information for multiresolution image registration," *IEEE Trans. Image Process.*, vol. 9, no. 12, pp. 2083–2099, Dec. 2000.
[13] S. Kabus, T. Netsch, B. Fischer, and J. Modersitzki, "B-spline registration of 3D images with Levenberg-Marquardt optimization," in *Med. Imag.*, 2004, pp. 304–313.
[14] M. Kisaki *et al.*, "High speed image registration of head CT and MR images based on Levenberg-Marquardt algorithms," in *Soft Comput. Intell. Syst., 2014 Joint 7th Int. Conf. Adv. Intell. Syst.*, 2014, pp. 1481–1485.
[15] 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. Imag.*, vol. 22, no. 1, pp. 120–128, Jan. 2003.
[16] M. Sdika, "A fast nonrigid image registration with constraints on the Jacobian using large scale constrained optimization," *IEEE Trans. Med. Imag.*, vol. 27, no. 2, pp. 271–281, Feb. 2008.
[17] S. Damas, O. Cordón, and J. Santamaría, "Medical image registration using evolutionary computation: An experimental survey," *IEEE Comput. Intell. Mag.*, vol. 6, no. 4, pp. 26–42, 2011.
[18] M. P. Wachowiak, R. Smolíková, Y. Zheng, J. M. Zurada, and A. S. Elmaghraby, "An approach to multimodal biomedical image registration utilizing particle swarm optimization," *IEEE Trans. Evolut. Comput.*, vol. 8, no. 3, pp. 289–301, Jun. 2004.
[19] Y.-W. Chen, C.-L. Lin, and A. Mimori, "Multimodal medical image registration using particle swarm optimization," in *Proc. 8th Int. Conf. Intell. Syst. Design Appl.*, 2008, vol. 3, pp. 127–131.
[20] A. A. Cole-Rhodes, K. L. Johnson, J. LeMoigne, and I. Zavorin, "Multiresolution registration of remote sensing imagery by optimization of mutual information using a stochastic gradient," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1495–1511, Dec. 2003.
[21] S. Klein, J. Pluim, M. Staring, and M. Viergever, "Adaptive stochastic gradient descent optimisation for image registration," *Int. J. Comput. Vis.*, vol. 81, no. 3, pp. 227–239, 2009.
[22] J. C. Spall, "Implementation of the simultaneous perturbation algorithm for stochastic optimization," *IEEE Trans. Aerospace Electron. Syst.*, vol. 34, no. 3, pp. 817–823, Jul. 1998.
[23] L. Bottou, "Stochastic gradient learning in neural networks," *Proc. Neuro-Nîmes*, vol. 91, no. 8, 1991.

[24] A. Harju, B. Barbiellini, S. Siljamäki, R. Nieminen, and G. Ortiz, "Stochastic gradient approximation: An efficient method to optimize many-body wave functions," *Phys. Rev. Lett.*, vol. 79, no. 7, p. 1173, 1997.

[25] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[26] R. Suri and Y. T. Leung, "Single run optimization of a SIMAN model for closed loop flexible assembly systems," in *Proc. 19th Conf. Winter Simulat.*, New York, 1987, pp. 738–748.

[27] R. Brennan and P. Rogers, "Stochastic optimization applied to a manufacturing system operation problem," in *Proc. Winter Simulat. Conf.*, Dec. 1995, pp. 857–864.

[28] H. Kesten, "Accelerated stochastic approximation," *Ann. Math. Stat.*, pp. 41–59, 1958.

[29] A. Gaivoronski, "Stochastic quasigradient methods and their implementation," *Numerical Tech. Stochastic Optimization*, vol. 10, pp. 313–351, 1988.

[30] Y.-H. Dai and H. Zhang, "Adaptive two-point stepsize gradient algorithm," *Numerical Algorithms*, vol. 27, no. 4, pp. 377–385, 2001.

[31] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. New York: Wiley, 2005, vol. 65.

[32] A. P. George and W. B. Powell, "Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming," *Mach. Learn.*, vol. 65, no. 1, pp. 167–198, 2006.

[33] R. Bhagalia, J. A. Fessler, and B. Kim, "Accelerated nonrigid intensity-based image registration using importance sampling," *IEEE Trans. Med. Imag.*, vol. 28, no. 8, pp. 1208–1216, Aug. 2009.

[34] Y. Qiao, B. Lelieveldt, and M. Staring, "Fast automatic estimation of the optimization step size for nonrigid image registration," in *SPIE Medical Imaging*, 2014, pp. 90 341A–90 341A.

[35] H. J. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. New York: Springer Science Business Media, 2003, vol. 35.

[36] A. Plakhov and P. Cruz, "A stochastic approximation algorithm with step-size adaptation," *J. Math. Sci.*, vol. 120, no. 1, pp. 964–973, 2004.

[37] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. Pluim, "elastix: A toolbox for intensity-based medical image registration," *IEEE Trans. Med. Imag.*, vol. 29, no. 1, pp. 196–205, Jan. 2010.

[38] D. Vysochanskij and Y. I. Petunin, "Justification of the 3σ rule for unimodal distributions," *Theory Probabil. Math. Stat.*, vol. 21, pp. 25–36, 1980.

[39] J. West *et al.*, "Comparison and evaluation of retrospective intermodality brain image registration techniques," *J. Comput. Assist. Tomogr.*, vol. 21, no. 4, pp. 554–568, 1997.

[40] J. Stolk *et al.*, "Progression parameters for emphysema: A clinical investigation," *Respiratory Med.*, vol. 101, no. 9, pp. 1924–1930, 2007.

[41] K. Murphy *et al.*, "Semi-automatic construction of reference standards for evaluation of image registration," *Med. Image Anal.*, vol. 15, no. 1, pp. 71–84, 2011.

[42] M. Staring *et al.*, "Towards local progression estimation of pulmonary emphysema using CT," *Med. Phys.*, vol. 41, no. 2, p. 021905, 2014.

[43] A. Hammers *et al.*, "Three-dimensional maximum probability atlas of the human brain, with particular reference to the temporal lobe," *Human Brain Mapp.*, vol. 19, no. 4, pp. 224–247, 2003.

[44] I. S. Gousias *et al.*, "Automatic segmentation of brain MRIs of 2-year-olds into 83 regions of interest," *NeuroImage*, vol. 40, no. 2, pp. 672–684, 2008.

[45] S. Vijayan *et al.*, "Motion tracking in the liver: Validation of a method based on 4D ultrasound using a nonrigid registration technique," *Med. Phys.*, vol. 41, no. 8, 2014.

[46] P. Seroul and D. Sarrut, "VV: A viewer for the evaluation of 4D image registration," in *Proc. MICCAI*, 2008, pp. 1–8.

[47] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE Trans. Med. Imag.*, vol. 16, no. 2, pp. 187–198, 1997.

[48] Life science grid [Online]. Available: https://surfsara.nl/project/life-science-grid

[49] D. Rueckert *et al.*, "Nonrigid registration using free-form deformations: Application to breast MR images," *IEEE Trans. Med. Imag.*, vol. 18, no. 8, pp. 712–721, Aug. 1999.

[50] C. Metz, S. Klein, M. Schaap, T. van Walsum, and W. J. Niessen, "Nonrigid registration of dynamic medical imaging data using $\mathrm{nD}+\mathrm{t}$ B-splines and a groupwise optimization approach," *Med. Image Anal.*, vol. 15, no. 2, pp. 238–249, 2011.

[51] W. Sun, W. Niessen, M. van Stralen, and S. Klein, "Simultaneous multiresolution strategies for nonrigid image registration," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4905–4917, Dec. 2013.

[52] N. N. Schraudolph and T. Graepel, "Combining conjugate direction methods with stochastic approximation of gradients," in *Proc. 9th Int. Workshop Artif. Intell. Stat.*, Jan. 2003.

[53] E. Laure *et al.*, Programming the Grid with gLite 2006 [Online]. Available: http://cds.cern.ch/record/936685

[54] P. Andreetto *et al.*, "The gLite workload management system," in *J. Phys. Conf. Ser.*, Jul. 2008, vol. 119, p. 062007, 6.

[55] EGI Site [Online]. Available: http://www.egi.eu/

[56] C. Marco, C. Fabio, D. Alvise, G. Antonia, G. Francesco, M. Alessandro, M. Moreno, M. Salvatore, P. Fabrizio, P. Luca, and P. Francesco, "The gLite workload management system," in *Advances in Grid and Pervasive Computing*, N. Abdennadher and D. Petcu, Eds. Berlin, Germany: Springer, 2009, vol. 5529, LNCS, pp. 256–268.

[57] A. Casajus, R. Graciani, S. Paterson, and A. Tsaregorodtsev, "DIRAC pilot framework and the DIRAC workload management system," in *J. Phys., Conf. Ser.*, Apr. 2010, vol. 219, p. 062049, 6.

[58] jjbot/picasclient [Online]. Available: https://github.com/jjbot/picasclient

[59] RP3/Grid Training | GitLab [Online]. Available: https://git.lumc.nl/rp3/grid_training/tree/1c654deb90d85a4c62cbc1cfac6f2fb64572a78b

[60] Python [Online]. Available: https://www.python.org/

[61] Ganga: Gaudi/Athena and Grid Alliance [Online]. Available: http://ganga.web.cern.ch/ganga/,