An Efficient Preconditioner for Stochastic Gradient Descent Optimization of Image Registration

Yuchuan Qiao, Boudewijn P. F. Lelieveldt, and Marius Staring

***Abstract*—Stochastic gradient descent (SGD) is commonly used to solve (parametric) image registration problems. In the case of badly scaled problems, SGD, however, only exhibits sublinear convergence properties. In this paper, we propose an efficient preconditioner estimation method to improve the convergence rate of SGD. Based on the observed distribution of voxel displacements in the registration, we estimate the diagonal entries of a preconditioning matrix, thus rescaling the optimization cost function. The preconditioner is efficient to compute and employ and can be used for mono-modal as well as multi-modal cost functions, in combination with different transformation models, such as the rigid, the affine, and the B-spline model. Experiments on different clinical datasets show that the proposed method, indeed, improves the convergence rate compared with SGD with speedups around 2∼5 in all tested settings while retaining the same level of registration accuracy.**

***Index Terms*—Optimization, preconditioning, stochastic gradient descent, image registration.**

I. Introduction

IMAGE registration is widely used in medical image analysis and has ample applications, e.g. in radiation therapy and segmentation [1]. This procedure can be used to align images from different modalities or different time points following a continuous deformation strategy. The strategy can be formulated as a (parametric) optimization problem to minimize the dissimilarity between a d-dimensional fixed image I_F and moving image I_M:

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

in which $\boldsymbol{x}$ is an image coordinate and $\boldsymbol{T}(\boldsymbol{x}, \boldsymbol{\mu})$ is a coordinate transformation parameterized by $\boldsymbol{\mu}$. For example, $\boldsymbol{\mu}$ consists of rotations and translations for a rigid transformation model, and control point displacements for a nonrigid transformation modeled by B-splines. For several clinical applications, for example online adaptive radiation therapy [2], image registration runtime is crucially important. In particular, online adaptive intensity-modulated proton therapy (IMPT) [3] is very sensitive to treatment-related uncertainties, such as patient set-up, inter-fraction and intra-fraction variations in the shape and position of the target volume and organs at risk. These uncertainties should be tackled at each treatment fraction by re-optimizing the treatment plan based on a new CT scan-of-the-day. Re-contouring of the daily CT scan can be done by propagating the contour from the planning CT scan according to the spatial correspondence obtained by image registration. The registration should be performed within the time span that new organ motion occurs (less than 30 seconds for the prostate [4]), especially when a small margin is applied. A computationally efficient optimization strategy for image registration, that yields high accuracies at the same time, is therefore required.

An iterative optimization scheme is typically used:

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

where k is the iteration number, γ_k is the step size at iteration k, and $\boldsymbol{D}_k$ is a search direction in the parameter space. Commonly used methods to determine the search direction $\boldsymbol{D}_k$ are of first order (gradient descent) or second order (Newton or quasi-Newton) descent type. Gradient descent, however, only achieves a sublinear convergence rate for nonconvex problems or a linear convergence rate for convex problems [5], [6]. Especially for badly scaled problems, these methods converge slowly. A common example of a badly scaled problem is a rigid registration where the translational parameters can have a magnitude in the order of 1-50 mm, while the rotational parameters typically have a magnitude $\ll 1$. Second order derivative methods such as the quasi-Newton method converge faster, however, the computation of the Hessian matrix update is very time consuming, especially when the number of image voxels and transformation parameters are large [7]. For registration problems with a

Manuscript received January 3, 2019; accepted February 3, 2019. Date of publication February 11, 2019; date of current version October 1, 2019. This work was supported in part by ZonMw, the Netherlands Organization for Health Research and Development, under Grant 104003012, and in part by the China Scholarship Council under Grant 201206130066. *(Corresponding author: Marius Staring.)*

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

B. P. F. Lelieveldt and M. Staring are with the Division of Image Processing, Department of Radiology, Leiden University Medical Center, 2300 RC Leiden, The Netherlands, and also with the Intelligent Systems Department, Faculty of EEMCS, Delft University of Technology, 2600 GA Delft, The Netherlands (e-mail: m.staring@lumc.nl).

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

Digital Object Identifier 10.1109/TMI.2019.2897943

large number of degrees of freedom and a large image size, it is not very efficient to calculate the search direction in a deterministic way [6] (i.e. using all voxels to compute the gradient). Klein *et al.* [8] proposed a stochastic gradient descent method for image registration, which approximates the gradient by only using a random subset of the image samples. This approximation is much more efficient to compute, thereby outperforming deterministic gradient descent and even quasi-Newton methods [6]. For badly scaled problems, however, SGD would suffer from a deteriorated convergence rate. To overcome these shortcomings, preconditioning techniques were proposed to turn a badly scaled optimization problem into a properly scaled one, considering the curvature of the cost function [5], [9]. The construction of these preconditioners can however be computationally expensive in themselves, which can easily mitigate the positive effect of faster convergence.

Two major groups of preconditioning techniques are widely used in iterative optimization. One, sometimes named variable preconditioning, uses the update rule: $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k - \gamma_k \boldsymbol{P}_k \boldsymbol{g}_k$. The preconditioner $\boldsymbol{P}_k$ is updated at each iteration (or at least regularly) to adapt to the local shape of the cost function [10]–[14]. This group of methods is typically used in machine learning to solve a linear system [10], [12], [15], [16], but is also popular in image registration [6], [17]. Popular preconditioners, such as Newton or quasi-Newton methods [14], indeed exhibit superior convergence rate compared to the standard gradient descent methods. These improvements, however, come at a cost of the estimation of the inverse Hessian, which alleviates some of the advantages and can even lead to a net deceleration. Zikic *et al.* [14] proposed a diagonal preconditioner for Demons registration. They applied the preconditioner to the dense gradient of the energy function using the inverse of the gradient magnitude. Besides its extra computational effort at each iteration, its performance mainly depends on the choice of a parameter ρ. This parameter is problem specific for different dissimilarity measures, different modalities and different transformation models, which may limit its practicality. Another group of preconditioning techniques, sometimes called traditional preconditioning, use a static $\boldsymbol{P}$, i.e. the preconditioner $\boldsymbol{P}$ is only calculated once before the start of the optimization [5], [9], [10]. The Krylov subspace method, sparse approximate inverse and Jacobi preconditioning techniques are often used [9]. Klein *et al.* [18] proposed a preconditioner construction method only suitable for certain cases of mono-modal image registration such as registration of 3D chest CT scans (of approximately the same breathing phase), which approximates the Hessian matrix of the cost function based on an assumption that the intensity difference between moving image and fixed image is zero after a perfect registration. This method is additionally very time-consuming when the number of transformation parameters and image size increase: the required decomposition of the Hessian matrix takes more than 3 hours for $\sim 10^5$ parameters with an image size of $450 \times 300 \times 150$ voxels using an Intel Xeon E5620 CPU with 8 cores running at 2.4 GHz.

In this paper we propose a novel fast preconditioned stochastic gradient descent method (FPSGD) for image registration. Based on a connection between the incremental displacement of a voxel and the gradient change between iterations, an efficient method to construct a diagonal preconditioner for stochastic gradient descent methods is derived. This diagonal preconditioner is different from traditional methods which utilize the Jacobian as a diagonal entry, and also does not rely on the Hessian matrix of the cost function [18]. Experimental results on four different datasets from different imaging modalities and different organs show a promising performance of the proposed method compared to the stochastic gradient descent method and other preconditioner estimation methods.

II. Background

A. Preconditioned Stochastic Gradient Descent

The PSGD is established as:

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k - \gamma_k \boldsymbol{P} \tilde{\boldsymbol{g}}_k, \tag{3}$$

where $\tilde{\boldsymbol{g}}_k$ is a stochastic gradient evaluated on a random subset of the image samples $\boldsymbol{\Omega}_F^s$ with a size of N_s and $\boldsymbol{P}$ is a positive definite $N_P \times N_P$ matrix, with N_P the number of parameters that model the transformation, i.e. $|\boldsymbol{\mu}|$. When $\boldsymbol{P} = \boldsymbol{I}$, PSGD will be reduced to the standard SGD method. The choice of $\boldsymbol{P} = \boldsymbol{H}^{-1}$ is another extreme where $\boldsymbol{H}^{-1}$ is the inverse Hessian of the cost function at the optimal parameter $\widehat{\boldsymbol{\mu}}$. Obviously, the calculation of the inverse Hessian has the same complexity as the original problem, and is not a time-efficient preconditioner. The convergence of PSGD is guaranteed when i) $\boldsymbol{P}$ is positive definite; and ii) the step size sequence is a non-increasing and non-zero sequence with $\sum_{k=1}^{\infty} \gamma_k = \infty$ and $\sum_{k=1}^{\infty} \gamma_k^2 < \infty$ [10], [11]. The step size sequence used here is defined as follows [18]:

$$\gamma_k = \begin{cases} 1 & \text{if } k = 0 \\ \dfrac{\eta}{(t_k + 1)/A + 1} & \text{if } k > 0 \end{cases}$$
$$t_k = \max(0, t_{k-1} + \mathrm{sigmoid}(-\tilde{\boldsymbol{g}}_{k-1}^T \boldsymbol{P} \tilde{\boldsymbol{g}}_{k-2})), \tag{4}$$

in which t_0 and t_1 equal to 0, η is a noise compensation factor and A controls the decay speed of the step size sequence and is typically set to 20. The noise introduced by the stochastic procedure will influence the convergence rate, so inspired from [18], [19] we use the following compensation factor:

$$\eta = \frac{E\|\boldsymbol{g}^T \boldsymbol{P} \boldsymbol{g}\|}{E\|\tilde{\boldsymbol{g}}^T \boldsymbol{P} \tilde{\boldsymbol{g}}\|} = \frac{E\|\boldsymbol{g}^T \boldsymbol{P} \boldsymbol{g}\|}{E\|\boldsymbol{g}^T \boldsymbol{P} \boldsymbol{g}\| + E\|\boldsymbol{\epsilon}^T \boldsymbol{P} \boldsymbol{\epsilon}\|}, \tag{5}$$

in which $\boldsymbol{g}$ is the exact gradient evaluated on all voxels in the image, $\boldsymbol{\epsilon}$ the random noise added to the exact gradient and $E\|\cdot\|$ is the expectation of the norm. For the (preconditioned) stochastic gradient descent method, the stop condition could be chosen as the moment when the exact cost function value does not decrease anymore. However, this would require an exact calculation over all image samples, which would take considerable time again. Therefore, the stop condition used in this paper is simply a maximum number of iterations K, as is typically used [8], [19]. The details of PSGD are given in Algorithm 1.

Algorithm 1 Preconditioned Stochastic Gradient Descent

```
Require: $N_s$ the number of samples for optimization, $\delta$ the maximum allowed voxel displacement, $K$ the number of iterations
for $k = 1, 2, \ldots, K$ do
    Randomly sample the whole image to get $N_s$ samples
    Calculate the gradient $\tilde{\boldsymbol{g}}_k$ over $N_s$ samples
    Calculate the stepsize $\gamma_k$ using Equation (4)
    Estimate the preconditioner $\boldsymbol{P}$
    Update the parameter $\boldsymbol{\mu}_{k+1}$ using Equation (3)
Return $\hat{\boldsymbol{\mu}}$
```

B. Related Work

There are several related works to estimate a preconditioner:

1) Hessian-type preconditioner (PSGD-H). The theoretical optimal choice for the preconditioner is the inverse Hessian at the optimal parameter $\widehat{\boldsymbol{\mu}}$. However, it is impossible to obtain the exact inverse Hessian beforehand because $\widehat{\boldsymbol{\mu}}$ is unknown [18]. Based on the assumption that the moving image is the same as the fixed image after successful registration: $\boldsymbol{F} \approx \boldsymbol{M}(\boldsymbol{T}(\boldsymbol{x}; \widehat{\boldsymbol{\mu}}))$, and the assumption that the deformation is small: $\partial \boldsymbol{T}/\partial \boldsymbol{\mu} \approx \boldsymbol{I}$, Klein *et al.* [18] proposed a method to approximate the Hessian-type preconditioner. However, a Cholesky decomposition is needed for calculating the preconditioner, with a computational complexity in the order of $\mathcal{O}(N_P^3)$. Therefore, the computation time of this preconditioner is very long when solving large scale problems, which nullifies the improvements in the convergence.

2) Jacobi-type preconditioner (PSGD-J). For rigid and affine registration problems, Klein *et al.* [18] assumed that the rotation parameters were scaled by the average voxel displacement caused by a small perturbation of the rotation angle, and proposed a method to construct a diagonal Jacobi-type preconditioner for PSGD. The elements p_i of the diagonal preconditioner $\boldsymbol{P}$ are calculated as follows:

$$p_i = \left(\int_{\Omega_F} \left\| \frac{\partial \boldsymbol{T}}{\partial \theta_i}(\boldsymbol{x}; \boldsymbol{\mu}_0) \right\|^2 \mathrm{d}x \Big/ \int_{\Omega_F} \mathrm{d}x \right)^{-\frac{1}{2}}. \tag{6}$$

The complexity of this method is $\mathcal{O}(N_P)$, which is very efficient. In this paper, we extend this method to non-linear registration problems using a B-spline parameterization.

3) AdaGrad [20] is a variable preconditioner estimation method well known from the machine learning field. This diagonal preconditioner is estimated as follows:

$$p_{k,i} = \frac{1}{\sqrt{\sum_{j=0}^{j=k} \tilde{g}_{j,i}^2 + \epsilon}}, \tag{7}$$

in which $\tilde{g}_{j,i}$ is i-th entry of the stochastic gradient at iteration j ($j \leq k$, with k the current iteration). The complexity is the same as the Jacobi-type preconditioner, i.e. $\mathcal{O}(N_P)$ for each iteration. Note that this preconditioner changes at each iteration making it a *variable* preconditioner, and that it becomes infinitesimally small as k increases.

III. Method

A. Preliminaries

The aim of preconditioning is to scale the parameter space so that the registration problem is easier to optimize. An ideal preconditioner should take care of the relative scaling between the parameters. Construction of a suitable preconditioner is a challenge for a given problem. First, different transformation models and different dissimilarity measures result in different characteristic of the cost function, making the determination of a preconditioner problem-specific. Second, the computation of the preconditioner should be efficient performance-wise, otherwise the overhead of the preconditioner computation will consume the advantage in runtime reductions obtained from the improvements of the convergence rate.

Inspired by our previous work [8, eq. (38)], [19, eq. (12)], we found that the incremental voxel displacement relates to the Jacobian of the transformation and the gradient of the cost function. Also note that it is easier to estimate as well as apply a diagonal preconditioner for image registration when we recall Equation (3). Different from [8] and [19] where only a scalar step size was proposed, here we construct a vector preconditioner $\boldsymbol{P} = \mathrm{diag}(\boldsymbol{p})$, with $\boldsymbol{P}$ of size N_P. In the following we will derive the i-th entry p_i of the preconditioner corresponding to the i-th entry of the transformation parameters $\boldsymbol{\mu}$, such that the displacement induced by a change in that parameter is equal to a predefined value δ. The incremental displacement of a voxel $\boldsymbol{x}_j$ in the fixed image domain Ω_F between iteration k and $k+1$ for an iterative optimization scheme is defined as:

$$\boldsymbol{d}_k(\boldsymbol{x}_j) = \boldsymbol{T}\left(\boldsymbol{x}_j, \boldsymbol{\mu}_{k+1}\right) - \boldsymbol{T}\left(\boldsymbol{x}_j, \boldsymbol{\mu}_k\right). \tag{8}$$

We approximate the incremental displacement $\boldsymbol{d}_k$ using the first-order Taylor expansion around $\boldsymbol{\mu}_k$:

$$\begin{aligned} \boldsymbol{d}_k(\boldsymbol{x}_j) &\approx \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{\mu}}\left(\boldsymbol{x}_j, \boldsymbol{\mu}_k\right) \cdot \left(\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k\right) \\ &= \boldsymbol{J}(\boldsymbol{x}_j) \cdot \left(\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k\right), \end{aligned} \tag{9}$$

in which $\boldsymbol{J}(\boldsymbol{x}_j) = \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{\mu}}\left(\boldsymbol{x}_j, \boldsymbol{\mu}_k\right)$ is the Jacobian matrix of size $d \times N_P$. All transformation models use the same derivations. Using the optimization scheme (3), we obtain $\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k = -\gamma_k \boldsymbol{P} \tilde{\boldsymbol{g}}_k$, and we can rewrite $\boldsymbol{d}_k$ as:

$$\boldsymbol{d}_k(\boldsymbol{x}_j) \approx -\gamma_k \boldsymbol{J}(\boldsymbol{x}_j) \boldsymbol{P} \tilde{\boldsymbol{g}}_k. \tag{10}$$

B. Diagonal Preconditioner Estimation

At iteration $k = 0$, i.e. prior to the start of the registration process, the preconditioner is estimated. From Equation (4) and Equation (10), we obtain $\boldsymbol{d}_0(\boldsymbol{x}_j) \approx -\boldsymbol{J}(\boldsymbol{x}_j)\mathrm{diag}(\boldsymbol{p})\tilde{\boldsymbol{g}}_0$. In the remainder of the paper, we use the notation $\boldsymbol{d}$ and $\tilde{\boldsymbol{g}}$ for simplification, instead of $\boldsymbol{d}_0$ and $\tilde{\boldsymbol{g}}_0$.

The Jacobi-type preconditioner from Equation (6) can be rewritten to:

$$p_i = \left(E \| \boldsymbol{J}^i(\boldsymbol{x}_j) \|^2 \right)^{-1/2}, \tag{11}$$

where $\boldsymbol{J}^i(\boldsymbol{x}_j)$ denotes the i-th column of the Jacobian matrix, and $\|\cdot\|$ is the ℓ_2 norm. Inspired by Equation (11), we can create a diagonal preconditioner but in a different form. We inspect the displacement $\|\boldsymbol{d}^i\|$ that is induced by a change $\triangle\mu_i$ in the i-th transformation parameter, i.e. the displacement generated by $\tilde{g}^i$ only:

$$\|\boldsymbol{d}^i(\boldsymbol{x}_j)\| \approx \left\|-\boldsymbol{J}^i(\boldsymbol{x}_j)p_i\tilde{g}^i\right\| = p_i \cdot \|\boldsymbol{J}^i(\boldsymbol{x}_j)\| \cdot \|\tilde{g}^i\|. \quad (12)$$

To constrain the voxel movement during the optimization, we assume that the voxel displacement $\boldsymbol{d}^i$ is to be not larger than δ: i.e $\|\boldsymbol{d}^i(\boldsymbol{x}_j)\| \leq \delta, \quad \forall \boldsymbol{x}_j \in \Omega_F$. In prior work [8] we found that as a rule of thumb setting δ to the mean voxel size gives satisfactory results, with the highest stable convergence rate. Based on the distribution of the voxel displacements, there is a weakened form for this assumption: $P(\|\boldsymbol{d}^i(\boldsymbol{x}_j)\| > \delta) < \rho$, where ρ is a small probability value often 0.05. According to the Vysochanskij-Petunin inequality [21], we have the following expression:

$$E\|\boldsymbol{d}^i(\boldsymbol{x}_j)\| + 2\sqrt{Var\|\boldsymbol{d}^i(\boldsymbol{x}_j)\|} \leq \delta, \quad \forall \boldsymbol{x}_j \in \Omega_F. \quad (13)$$

Combined with Equation (12), we obtain the relationship between the i-th entry p_i of the preconditioner and the maximum voxel displacement as follows:

$$p_i\left(E(s_i(\boldsymbol{x}_j)) + 2\sqrt{Var(s_i(\boldsymbol{x}_j))}\right) \leq \delta, \quad (14)$$

where $s_i(\boldsymbol{x}_j) = \|\boldsymbol{J}^i(\boldsymbol{x}_j)\| \cdot \|\tilde{g}^i\|$. The i-th entry of the preconditioner is then defined as:

$$p_i = \frac{\delta}{E(s_i(\boldsymbol{x}_j)) + 2\sqrt{Var(s_i(\boldsymbol{x}_j))} + \varepsilon}, \quad (15)$$

where ε is a small number to avoid division by zero. Finally, the full preconditioner $\boldsymbol{P}$ is obtained by repeating the above procedure for each p_i. Note that the number of samples N_{sp} for preconditioner estimation is not equal to the total number of voxels in the fixed image, but only a subset of all voxels for computational efficiency.

C. Regularization

The assumption used to approximate a preconditioner, that all transformation parameters should independently induce a maximum voxel displacement δ, may be too strict or too sensitive to noise in the measurements. For the B-spline transformation, for example, this assumption forces all regions to have a displacement δ, even regions that do not require registration. Noise could come from an insufficient number of samples $\boldsymbol{x}_j$ used for the estimation, or from inexact evaluation of the gradient. This could result in differences in the estimated entries of the preconditioner that are expected to have similar value. For the B-spline transformation model one would expect that nearby control points would be scaled similarly, without sudden sharp transitions. For the affine transformation on the other hand, one would expect that scalings related to translation parameters are more similar than those related to rotational parameters. We therefore propose to optionally regularize the procedure from Section III-B, such that the i-th entry p_i of the preconditioner is not treated completely independent, but also takes into account the estimates of the related parameters. Related parameters are those jointly affected by a voxel $\boldsymbol{x}_j$ (for an affine transformation these are all parameters; for the B-spline only parameters in the compact support region of $\boldsymbol{x}_j$), and secondly by their similarity in Jacobian contribution (for the affine transformation, intuitively rotations and translations are to be treated separately). The proposed regularization procedure is as follows:

$$s_i(\boldsymbol{x}_j) = \tau \cdot s_i(\boldsymbol{x}_j) + \underbrace{\frac{1-\tau}{\sum\omega_m}\sum_{m\neq i} s_m(\boldsymbol{x}_j)\cdot\omega_m}_{\text{regularization term}}, \quad (16)$$

where ω_m weighs the contributions of similar parameters and τ balances the contribution of entry i with the contributions of the other parameters. We expect the weights ω_m to hold the property that a large weight is taken for similar Jacobian terms and a small weight for dissimilar Jacobian terms (a larger difference). Therefore, the weights ω_m are chosen using a Gaussian function:

$$\omega_m = \exp\left(-\frac{(\|\boldsymbol{J}^i(\boldsymbol{x}_j)\| - \|\boldsymbol{J}^m(\boldsymbol{x}_j)\|)^2}{2\sigma^2}\right), \quad (17)$$

in which σ is chosen as

$$\sigma = \frac{\min(\|\boldsymbol{J}^i(\boldsymbol{x}_j)\| - \|\boldsymbol{J}^m(\boldsymbol{x}_j)\|)}{\max(\|\boldsymbol{J}^i(\boldsymbol{x}_j)\| - \|\boldsymbol{J}^m(\boldsymbol{x}_j)\|)}, \quad \forall m \neq i. \quad (18)$$

Note that ω_m weighs the difference between rotation and translation. The selection of σ for the Gaussian function is based on the observed range of Jacobian values for normalization. While for the B-spline transformation model such a choice would also be valid, a simplification is possible. For the B-spline model the displacement of a voxel is only determined by the control points in its support region. Furthermore, we expect the influence on the displacement to be almost equal for each control point in the support region. We therefore assume for the B-spline model that the weights $\omega_m = 1$, simplifying Equation (16) to $s_i(\boldsymbol{x}_j) = \tau\cdot s_i(\boldsymbol{x}_j) + (1-\tau)\cdot\sum_{m\neq i} s_m(\boldsymbol{x}_j)/N_{cp}$, where N_{cp} is the number of control points.

D. Condition Number

Until now, we efficiently constructed a diagonal preconditioner through the estimation from the Jacobian and the gradient. However, the condition number of the estimated preconditioner matrix $\boldsymbol{P}$ may still be high, which may influence the stability of the optimization procedure. In the following, we constrain the condition number of the preconditioner matrix, to make the optimization procedure more robust and stable [5], [9], [18]. The convergence rate of the registration algorithm can be measured by the so-called condition number: $\kappa = \lambda_{\max}/\lambda_{\min}$, where $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalue of $\boldsymbol{P}$, respectively. It is common to constrain the eigenvalues, such that the condition number will be closer to 1 [13], [18]. We introduce a user-defined maximum condition number $\kappa_{\max}$ for this purpose.

Define a diagonal eigenvalue matrix $\Lambda = diag(\lambda_1, \ldots, \lambda_{N_p})$ for the preconditioner $\boldsymbol{P}$. In this study, as our preconditioner $\boldsymbol{P}$ is diagonal, the entries of $\boldsymbol{P}$ are equal to the eigenvalues of Λ: $p_i = \lambda_i, \forall i$. To constrain the eigenvalues, we replace small eigenvalues of $\boldsymbol{P}$ that make $\kappa > \kappa_{\max}$ using the following equation:

$$p_i = \begin{cases} \lambda_{\max}/\kappa_{\max}, & \text{if } \lambda_{\max}/\lambda_i > \kappa_{\max}, \\ \lambda_i, & \text{otherwise.} \end{cases} \tag{19}$$

Then the constrained matrix from Equation (19) constitutes the finally proposed static preconditioner.

E. Summary and Complexity Analysis

In summary, using $\boldsymbol{P}$ as defined in Equation (15), (16) and (19), and γ_k as in Equation (4), Equation (3) results in the Fast Preconditioned Stochastic Gradient Descent method (FPSGD). The procedure is detailed in Algorithm 2.

Algorithm 2 Proposed Preconditioner Estimation

```
Require: $N_{sp}$ the number of samples for preconditioner estimation, $\delta$ the maximum allowed voxel displacement, $\tau$ the
    regularization factor, $\kappa_{\max}$ the maximum condition number
1: Compute the gradient $\tilde{\boldsymbol{g}}$ on $N_{sp}$ random samples
2: Randomly take $N_{sJ}$ samples $\{\boldsymbol{x}_j\}$ from the fixed image
3: $\boldsymbol{p} = \boldsymbol{I}$, $\boldsymbol{t} = \boldsymbol{0}$, $\boldsymbol{z} = \boldsymbol{0}$, $\boldsymbol{y} = \boldsymbol{0}$    ▷ initialization
4: for $j = 1, 2, \ldots, N_{sJ}$ do    ▷ loop over the samples $\boldsymbol{x}_j$
5:     Calculate the Jacobian $\boldsymbol{J}(\boldsymbol{x}_j)$
6:     for $i = 1, 2, \ldots, N_P$ do    ▷ loop over the parameters
7:         $s_i = \|\boldsymbol{J}^i(\boldsymbol{x}_j)\| \cdot \|\tilde{g}^i\|$
8:         Regularize $s_i$ with $\tau$ using Section III-C
9:         $z_i = z_i + s_i$    ▷ update for the mean
10:        $y_i = y_i + s_i^2$    ▷ update for the variance
11:        $c_i = c_i + 1$    ▷ increase counter
12: for $i = 1, 2, \ldots, N_P$ do    ▷ loop over the parameters
13:     $q_i = z_i/c_i + 2\sqrt{(y_i/c_i) - (z_i/c_i)^2}$
14:     $p_i = \delta/(q_i + \varepsilon)$
15:     Constrain the condition number of $p_i$ using $\kappa_{\max}$ (see
        Section III-D)
16: Return $\boldsymbol{p}$
```

As we can see from Equation (15), each entry p_i of the proposed preconditioner is positive. Since the proposed preconditioner is also diagonal and thus symmetric, it is easy to derive that $\boldsymbol{P}$ is positive definite. $\boldsymbol{P}$ thus fulfills the necessary conditions for a valid preconditioner, stated in Section II-A and [5], [9]. After the condition number modification in Equation (19), the proposed preconditioner makes the iterative optimization more robust [5], [9], [18].

From Algorithm 2, we derive the time complexity of the proposed method as follows:

- In step 1, the gradient $\tilde{\boldsymbol{g}}$ is computed using N_{sp} samples. Assuming that the cost function derivative for a single sample has a constant and relatively low complexity, the computational complexity of this step is $\mathcal{O}(N_{sp})$.
- In step 4 to step 11, the Jacobian matrix $\boldsymbol{J}(\boldsymbol{x})$ of size $d \times N_P$ is computed using N_{sJ} samples. Assuming that the derivative of the transformation for a single sample and a single parameter again has a constant and relatively low complexity, the computational complexity of step 7 is $\mathcal{O}(N_{sJ} \times N_P)$. The regularization term loops over all other entries of the preconditioner and therefore has a time complexity of $\mathcal{O}(N_{sJ} \times N_P \times N_P)$.
- From step 12 to 15, the time complexity of the condition number modification is $\mathcal{O}(N_P)$.
- The total time complexity of the proposed FPSGD method is then $\mathcal{O}(N_{sp} + N_{sJ} \times N_P + N_{sJ} \times N_P \times N_P + N_P)$. For rigid and affine transformations, N_P and N_{sJ} are much smaller than N_{sp}, so the time complexity reduces to $\mathcal{O}(N_{sp})$, i.e. dominated by the number of samples. For the B-spline transformation, due to the compact support, there are only N_{cp} non-zero entries for each sample. Then, the Jacobian calculation is $\mathcal{O}(N_{sJ} N_{cp})$, and the regularization $\mathcal{O}(N_{sJ} N_P N_{cp})$. In total for the B-spline transformation, given that $N_P \gg N_{cp}$, the time complexity is $\mathcal{O}(N_{sJ} N_P)$.

The PSGD-H method has a time complexity of $\mathcal{O}(N_P^3)$, so for the same number of samples, the time complexity of the proposed method is linear with respect to the number of transformation parameters instead of a power of 3. The FASGD and the proposed method both employ a linear time complexity with respect to the number of parameters, whereas empirically FASGD is about twice as fast to compute than the proposed FPSGD.

IV. Data Sets

A. Mono-Modal Data: SPREAD Lung CT Data

3D lung Computed Tomography (CT) images of 19 patients were acquired during the SPREAD study [22]. A follow-up scan was acquired for each patient after the baseline scan with image sizes around 450 × 300 × 150 and voxel sizes around 0.7 × 0.7 × 2.5 mm. The ground truth consists of 100 anatomical corresponding points, which were semi-automatically extracted using Murphy's method [23]. The algorithm first automatically selects 100 evenly distributed landmarks within the pre-segmented lungs at characteristic locations in the baseline image, and then predicts the corresponding points in the follow-up image. The corresponding points are then inspected and corrected by two experts using a graphical user interface.

B. Mono-Modal Data: Prostate CT Data

CT images of 18 patients treated for prostate cancer with intensity-modulated radiation therapy were scanned at Haukeland University Hospital in 2007 [3]. For each patient, a planning CT image and 7-10 repeat CT images were acquired out-of-room, resulting in a total of 179 CT images. Each CT image contained 90 to 180 slices and had a slice thickness of 2-3 mm. Each slice had an in-slice pixel resolution in the range from 0.84 × 0.84 mm to 0.95 × 0.95 mm and totally 512 × 512 pixels. For each CT image, the prostate was delineated by an expert, and independently reviewed by another expert [3], and these delineations serve as a ground truth for the evaluation.

C. Multi-Modal Brain Data: RIRE

This brain dataset was acquired during the Retrospective Image Registration Evaluation (RIRE) project. CT scans and Magnetic Resonance Imaging (MRI-T1) are available for 9 patients. The CT images have sizes of $512 \times 512 \times 50$ with voxel sizes of $0.45 \times 0.45 \times 3$ mm, while the MRI-T1 image is of size $256 \times 256 \times 50$ with voxel sizes of $0.85 \times 0.85 \times 3$ mm. Fiducial markers were implanted in each patient and served as a ground truth [24]. These markers were manually erased from the images and replaced with a simulated background pattern.

D. Multi-Modal Brain Data: BrainWeb Simulated Data

T1 and T2 weighted 3D brain MR images were created using the Simulated Brain Database from BrainWeb [25]. To generate brain image pairs, default settings provided by BrainWeb were used with 3% noise and 20% intensity non-uniformity. The brain images are of sizes $181 \times 217 \times 181$ and a voxel spacing of 1 mm isotropically. A mask of the brain was extracted from the T1 image by FSL-BET [26] and the same mask was used for the T2 image. 100 randomly generated displacement vector fields (DVFs) serve as the ground truth deformation fields. The DVFs are isotropically generated in three dimensions within the brain mask and the maximum magnitude of DVFs is chosen as 5, 8, 10 and 15 mm. These DVFs are then smoothed by a Gaussian filter with a standard deviation between 5 and 40 mm.

V. Experiments

In this section, experimental settings are given to test the performance of the proposed method. The proposed FPSGD method is compared with the following methods:

1) Fast adaptive stochastic gradient descent (FASGD) [19], which is a state-of-the-art first order stochastic optimization method that does not use preconditioning. For rigid and affine registration, the diagonal of the preconditioner $\boldsymbol{P}$ is chosen as 1 for the translational parameters and 1/100000 for the others. This reflects that the parameters $\boldsymbol{\mu}$ corresponding to rotation have in general a much smaller range than parameters corresponding to translation. This choice is the default setting of `elastix` [27], shown to work well in practice.
2) Jacobi-type preconditioner (PSGD-J) [18], where a diagonal preconditioner is chosen according to Equation (6). The stepsize is automatically estimated using the method provided in [19].
3) Hessian-type preconditioner (PSGD-H) [18], see Section II-B. This preconditioner is only suitable for mono-modal registration, and therefore only implemented for the mean squared intensity difference (MSD) dissimilarity measure.
4) AdaGrad [20], see Section II-B, which is a variable preconditioner estimation method well known from the machine learning field.

All these methods, including the proposed method, were implemented in C++ and are available as open source software via the `elastix` package. All methods were implemented using data parallelism by processing the sampled image voxels concurrently. All experiments were performed on a workstation with an Ubuntu Linux OS, which has 12 cores running at 3.6 GHz and 64 GB of memory. All experiments are carried out in multi-threaded mode. Detailed settings are presented in Section V-A.

A. Experimental Setup

To validate the generality of the proposed preconditioner, the experiments are performed on mono-modal as well as multi-modal image registration. For each group, different transformation models are used, namely the rigid, affine and B-spline transformation models [27]. For rigid and affine image registration, only one resolution of 500 iterations is used, to be able to more easily compare convergence properties. Adding more resolutions would yield different optimization starting points at later resolutions, making this comparison hard. For B-spline image registration, a realistic three-level multi-resolution framework is used on the SPREAD and prostate CT data with a standard deviation of the Gaussian smoothing filter of 2, 1 and 0.5 mm, and 500 iterations for each resolution. For the BrainWeb data, we used only one resolution for B-spline registration, where one resolution is easier for comparing convergence rates between methods than when using multiple resolutions. 1000 iterations were taken to ensure the registration converges.

The number of samples N_s used for computing $\tilde{\boldsymbol{g}}_k$ was the same for all methods and set to 5,000 [19]. Different methods used different number of samples N_{sp} for the preconditioner estimation or step size estimation. The influence of the number of samples for the preconditioner estimation was tested for the proposed FPSGD method (see Section VI-A.3), and in the remainder experiments, N_{sp} was chosen as 50,000 for FPSGD. For PSGD-J and FASGD, N_{sp} is 5,000 for each resolution. For PSGD-H the number of samples N_{sp} were set to 100,000 in resolution 1 and 2, and 500,000 in resolution 3, according to previous study [18]. The number of samples N_{sJ} has a same setting for FASGD, PSGD-J and the proposed FPSGD method, which was chosen equal to the number of transformation parameters N_P at each resolution, while 1000 samples were chosen when rigid or affine transformation was applied. For instance in the SPREAD experiment N_P is around 4,000, 15,000 and 90,000 samples for the three resolutions, respectively. The user pre-defined value δ for FASGD and the proposed FPSGD method is chosen as the mean length of the voxel size. $A = 20$ is used for all tested methods.

In Section III, there are two free parameters which would affect the performance of the proposed FPSGD method: the regularization factor τ and the maximum condition number $\kappa_{\max}$. To assess the influence of these two parameters, we first vary the regularization factor τ using a fixed $\kappa_{\max}$, and then vice versa. The regularization factor τ was selected between 0 and 1, using increments of 0.2, so there were 6 variations. For these tests, $\kappa_{\max} = 2$ was chosen for the B-spline registration, while for rigid and affine registration no restriction

TABLE I

The Influence of the Regularization Parameter τ in Affine Registration for The spread Lung CT Data, for the Proposed FPSGD Method. We Used the MSD Dissimilarity Measure, 1 Resolution, 500 Iterations and $\kappa_{\text{MAX}} = \infty$. Note That 1 Registration Failed for FPSGD $\tau = 1.0$

Optimizer	Iterations I avg ± std	t_{est} (s) avg ± std	t_{pure} (s) avg ± std	t_{total} (s) avg ± std	Speed-up avg ± std	ED (mm) avg ± std
FASGD	484 ± 15	0.05 ± 0.01	1.40 ± 0.11	1.45 ± 0.11	-	4.98 ± 3.45
FPSGD $\tau = 0.0$	79 ± 83	0.16 ± 0.02	0.17 ± 0.15	0.33 ± 0.15	5.1 ± 1.6	4.51 ± 3.25
FPSGD $\tau = 0.2$	43 ± 35	0.16 ± 0.03	0.10 ± 0.07	0.27 ± 0.08	5.8 ± 1.5	4.48 ± 3.27
FPSGD $\tau = 0.4$	38 ± 23	0.15 ± 0.03	0.09 ± 0.05	0.24 ± 0.06	6.4 ± 1.4	4.48 ± 3.26
FPSGD $\tau = 0.6$	34 ± 18	0.16 ± 0.03	0.08 ± 0.04	0.24 ± 0.05	6.3 ± 1.6	4.47 ± 3.22
FPSGD $\tau = 0.8$	42 ± 27	0.16 ± 0.03	0.10 ± 0.06	0.26 ± 0.06	5.8 ± 1.4	4.49 ± 3.25
FPSGD $\tau = 1.0$	79 ± 83	0.16 ± 0.03	0.16 ± 0.16	0.32 ± 0.16	5.0 ± 1.8	4.52 ± 3.18

is needed on the condition number, i.e. $\kappa_{\max} = \infty$. In the second group of tests, a fixed $\tau = 0.6$ was chosen and $\kappa_{\max} \in \{1, 2, 4, 8, 16\}$ were tested for the B-spline registrations of the SPREAD data and the BrainWeb data. The results are reported in Section VI-A.

B. Convergence and Runtime Performance

The performance of the tested methods is first evaluated in terms of the convergence rate and the resulting speed-up in runtime. To measure the convergence rate, the dissimilarity measure (MSD or MI) was calculated at each 5th iteration. This calculation was performed deterministically using all samples from the fixed image. FASGD is chosen as the baseline method and we compare the exact cost function value of all other methods against the exact cost function value of FASGD at its final solution $\widehat{\boldsymbol{\mu}}_{\text{ref}}$. For each method, we counted the number of iterations I required to obtain a cost function value that is equal to or smaller than that of the baseline method using $\mathcal{C}(\boldsymbol{\mu}_k) \leq \mathcal{C}(\widehat{\boldsymbol{\mu}}_{\text{ref}})$ for the first time.

To assess runtime performance, several computations are timed and recorded: the time t_{est} it takes to estimate the preconditioner $\boldsymbol{P}$ and the time t_{iter} each iteration takes. When I equals the number of iterations needed for reaching the same cost function value as FASGD, then the pure registration time is defined as $t_{\text{pure}} = t_{\text{iter}} \cdot I$. The total registration time is then $t_{\text{total}} = t_{\text{est}} + t_{\text{pure}}$. The time t_{est} consists of the time to estimate the preconditioner and/or the step size γ_0 for the different methods, i.e. for FASGD t_{est} is the estimation time of the step size, for PSGD-J and PSGD-H both are included and for the proposed FPSGD method t_{est} is the estimation time of the preconditioner.

C. Evaluation Measures

We used three measures to evaluate the registration accuracy, namely the Euclidean distance (ED) of corresponding points, the Dice similarity coefficient (DSC) of manual delineations, and the average residual deformation of random initial deformations. All registration accuracy results were analyzed with the Wilcoxon signed-rank test to evaluate statistical differences of these methods compared to the FASGD method.

The Euclidean distance between corresponding points is computed using ED $= \frac{1}{Z_p} \sum_{i=1}^{Z_p} \|\boldsymbol{T}_{\widehat{\boldsymbol{\mu}}}(\boldsymbol{p}_F^i) - \boldsymbol{p}_M^i\|$, with $\boldsymbol{p}_F$ and $\boldsymbol{p}_M$ the corresponding points, and $\boldsymbol{T}$ the transformation at iteration I. Z_p is the number of corresponding points, which is 100 and 8 for SPREAD lung CT data and RIRE brain data, respectively. This measure was used for SPREAD data and RIRE brain data, which had 100 corresponding points and 8 corner points, respectively. The DSC was chosen as a quantitative evaluation measure of the registration accuracy for the delineated prostate region: DSC $= \frac{2|\boldsymbol{R}_M \cap \boldsymbol{R}_F|}{|\boldsymbol{R}_M| + |\boldsymbol{R}_F|}$, where $\boldsymbol{R}_F$ and $\boldsymbol{R}_M$ are the manually delineated region in the repeat CT scan and the propagated region in the planning CT scan, respectively. The average residual deformation inside the brain mask $\boldsymbol{\Omega}_F$ was used to measure the recovery performance of initial deformation $\boldsymbol{T}_{\text{init}}$ for BrainWeb data: $Resi(\boldsymbol{T}_{\text{init}}, \boldsymbol{T}_{\widehat{\boldsymbol{\mu}}}) = \frac{1}{|\Omega_F|} \sum_{\boldsymbol{x}_i \in \Omega_F} \|\boldsymbol{T}_{\widehat{\boldsymbol{\mu}}}(\boldsymbol{T}_{\text{init}}(\boldsymbol{x}_i) - \boldsymbol{x}_i)\|$.

TABLE II

The Influence of the Regularization Parameter τ in B-Spline Registration for the SPREAD Lung CT Data, for the Proposed FPSGD Method. We Used the MSD Dissimilarity Measure, 3 Resolutions, 500 Iterations, and $\kappa_{\text{MAX}} = 4$

Optimizer	Res 1 Iterations avg ± std	Res 2 Iterations avg ± std	Res 3 Iterations avg ± std	ED (mm) avg ± std
FASGD	496 ± 0	496 ± 0	493 ± 5	1.66 ± 1.72
FPSGD $\tau = 0.0$	416 ± 73	305 ± 86	349 ± 106	1.63 ± 1.68
FPSGD $\tau = 0.2$	357 ± 75	243 ± 78	306 ± 114	1.60 ± 1.64
FPSGD $\tau = 0.4$	297 ± 70	186 ± 61	240 ± 108	1.57 ± 1.61
FPSGD $\tau = 0.6$	232 ± 69	133 ± 32	197 ± 89	1.53 ± 1.54
FPSGD $\tau = 0.8$	168 ± 53	118 ± 52	202 ± 138	1.47 ± 1.43
FPSGD $\tau = 1.0$	104 ± 50	191 ± 150	230 ± 153	1.43 ± 1.37

VI. Results

A. Parameter Sensitivity Analysis

In this section we evaluate the influence of several important parameters on SPREAD lung CT data: the regularization factor τ, the condition number $\kappa_{\max}$ and the number of samples N_{sp}. After the evaluation, we chose the optimal parameters for the remainder of the paper.

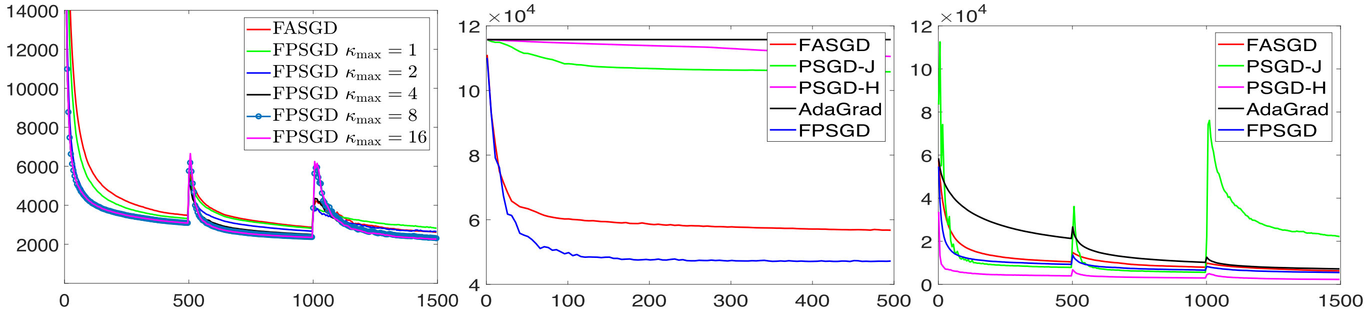
1) Regularization Factor τ:

Here we evaluate the influence of the parameter τ, using the SPREAD data. The results can be found in Table I and Table II. It can be seen from Table I that the regularization factor $\tau = 1.0$ (no regularization) gave the worst performance for affine registration, and in some cases resulted in failed registrations. Setting the regularization factor $\tau = 0.0$ is another extreme meaning that the regularization term completely determines the estimation of the preconditioner. From the B-spline results in Table II, it can be seen that the convergence rate is much slower for $\tau = 0$ than for

TABLE III

THE INFLUENCE OF κ_{MAX} ON B-SPLINE REGISTRATION ON THE SPREAD LUNG CT DATA, FOR THE PROPOSED FPSGD METHOD. WE USED THE MI DISSIMILARITY MEASURE, 3 RESOLUTIONS, 500 ITERATIONS, AND $\tau = 0.6$

Optimizer	Resolution 1 Iterations I avg ± std	Resolution 2 Iterations I avg ± std	Resolution 3 Iterations I avg ± std	ED (mm) avg ± std	p-value
FASGD	496 ± 0	496 ± 0	493 ± 5	1.66 ± 1.72	-
FPSGD $\kappa_{\max} = 1$	421 ± 62	405 ± 88	461 ± 80	1.70 ± 1.73	0.0017
FPSGD $\kappa_{\max} = 2$	282 ± 60	213 ± 62	293 ± 117	1.62 ± 1.64	0.0766
FPSGD $\kappa_{\max} = 4$	232 ± 69	133 ± 32	197 ± 89	1.53 ± 1.54	0.0002
FPSGD $\kappa_{\max} = 8$	227 ± 73	124 ± 43	190 ± 123	1.45 ± 1.46	0.0001
FPSGD $\kappa_{\max} = 16$	226 ± 72	121 ± 44	202 ± 147	1.45 ± 1.46	0.0006



(a) (b) (c)

Fig. 1. Convergence plots in the experiments of the SPREAD lung CT data, showing the cost function value (MSD) against the iteration number. (a) Example of $\kappa_{\max}$ difference. (b) Example of affine registration. (c) Example of B-spline registration.

TABLE IV

THE INFLUENCE OF THE NUMBER OF SAMPLES N_{sp} ON THE SPREAD LUNG CT DATA, FOR THE PROPOSED FPSGD METHOD. WE USED THE MSD DISSIMILARITY MEASURE, B-SPLINE TRANSFORMATION, 3 RESOLUTIONS, 500 ITERATIONS, $\kappa_{\text{MAX}} = 4$ AND $\tau = 0.6$. K INDICATES THE NUMBER OF THOUSANDS

	Resolution 1		Resolution 2		Resolution 3			
Optimizer	Iters I avg ± std	$t_{\text{est}}(s)$ avg ± std	Iters I avg ± std	$t_{\text{est}}(s)$ avg ± std	Iters I avg ± std	$t_{\text{est}}(s)$ avg ± std	ED (mm) avg ± std	p-value
FASGD	496 ± 0	0.16 ± 0.04	496 ± 0	0.09 ± 0.01	493 ± 5	0.15 ± 0.03	1.66 ± 1.72	-
FPSGD $N_{sp} = 5K$	264 ± 95	0.22 ± 0.03	265 ± 104	0.32 ± 0.05	403 ± 138	0.93 ± 0.16	1.66 ± 1.70	0.968
FPSGD $N_{sp} = 10K$	246 ± 62	0.23 ± 0.03	215 ± 81	0.34 ± 0.05	337 ± 115	0.93 ± 0.16	1.63 ± 1.67	0.198
FPSGD $N_{sp} = 20K$	238 ± 72	0.24 ± 0.03	164 ± 59	0.33 ± 0.06	232 ± 100	0.93 ± 0.17	1.58 ± 1.61	< 0.001
FPSGD $N_{sp} = 50K$	232 ± 69	0.28 ± 0.04	133 ± 32	0.34 ± 0.04	197 ± 89	0.96 ± 0.16	1.53 ± 1.54	< 0.001
FPSGD $N_{sp}=100K$	229 ± 68	0.34 ± 0.03	129 ± 40	0.38 ± 0.05	197 ± 115	1.01 ± 0.17	1.49 ± 1.49	< 0.001

other choices of τ, even though the registration accuracy is almost similar. The experimental results show that there was no statistical difference ($p < 0.001$) between the different choices of τ ($0.0 < \tau < 1.0$) regarding the accuracy. This reflects that the proposed preconditioner estimation method is quite robust to the selected τ in this application with respect to registration accuracy. With respect to the convergence rate larger values of τ give better results. We therefore conclude that a regularization factor τ between 0.6 and 0.8 gives the best overall results. In the remainder of the paper we use $\tau = 0.6$.

2) The Condition Number κ_{max}: The maximum condition number $\kappa_{\max}$ is especially important for non-rigid registration. Table III presents the registration accuracy with respect to $\kappa_{\max}$ for the SPREAD study. As we can see, different $\kappa_{\max}$ obtained a similar accuracy. However, fewer iterations were needed for a larger $\kappa_{\max}$. From the convergence plot in Figure 1a, it can be observed that the optimization converged faster for $\kappa_{\max} \geq 2$. However, for $\kappa_{\max} \geq 8$, the plot exhibits more oscillating behavior, suggesting a less stable optimization. In the remainder of the paper we set $\kappa_{\max} = 4$ for B-spline registration (and $\kappa_{\max} = \infty$ for rigid and affine registration).

3) The Number of Samples N_{sp}: To reduce the computation time of preconditioner estimation, we select a subset of samples from all image voxels. This approach would inherently influence the estimation accuracy and speed. To validate this influence, we performed an experiment with a varying number of samples. The experiment was performed on SPREAD lung CT data, with MSD and the B-spline transformation model. Three resolutions and 500 iterations at each resolution are applied. The results of the influence of the number of samples are given in Table IV. It can be clearly seen that the estimation time increases when increasing N_{sp}. The accuracy improves for larger N_{sp}, but only slightly. In summary, $N_{sp} = 50,000$ samples gives a good trade-off between estimation time and accuracy, and is chosen in the remainder of this paper.

B. Comparison of Different Methods

In this section we compare different preconditioner estimation methods on four datasets: SPREAD lung CT data, prostate CT data, RIRE brain data and BrainWeb simulated data.

TABLE V

Method Comparison for Affine Registration on the SPREAD Lung CT Data. We Used the MSD Dissimilarity Measure, 1 Resolution, 500 Iterations, $\tau = 0.6$ and $\kappa_{\text{MAX}} = \infty$

Optimizer	Iterations I	t_{est} (s)	t_{pure} (s)	t_{total} (s)	Speed-up	ED (mm)	p-value
	avg ± std	avg ± std	avg ± std	avg ± std	avg ± std	avg ± std	
FASGD	484 ± 15	0.05 ± 0.01	1.40 ± 0.11	1.45 ± 0.11	-	4.98 ± 3.45	-
PSGD-J	496 ± 0	0.07 ± 0.01	0.85 ± 0.18	0.93 ± 0.19	1.6 ± 0.2	10.1 ± 5.68	< 0.001
PSGD-H	496 ± 0	0.70 ± 0.14	0.83 ± 0.04	1.53 ± 0.17	1.0 ± 0.1	10.1 ± 5.62	< 0.001
AdaGrad	496 ± 0	0.05 ± 0.01	0.84 ± 0.11	0.90 ± 0.11	1.6 ± 0.2	10.8 ± 5.67	< 0.001
FPSGD	34 ± 18	0.17 ± 0.03	0.08 ± 0.04	0.25 ± 0.05	5.9 ± 1.4	4.47 ± 3.22	< 0.001

1) SPREAD Lung CT Data: The overall results of the experiments on affine registration for the SPREAD lung CT data comparing the different methods are given in Table V. It shows that the proposed FPSGD method took fewer iterations to obtain the same cost function value $\mathcal{C}(\widehat{\boldsymbol{\mu}}_{\text{ref}})$ than FASGD and PSGD-J. The speed-up in terms of number of iterations of the proposed FPSGD method is about 10. The improvements of the proposed FPSGD method compared to FASGD and PSGD-J in the convergence rate are also shown in Figure 1b. PSGD-H required fewer iterations than the proposed FPSGD method. The computation of the preconditioner however took somewhat longer, because the self-Hessian is calculated at each voxel and the number of samples used for the self-Hessian is larger than for the other methods, resulting in an overall decrease in performance. The overall speed-up in terms of runtime is about 5 for the proposed FPSGD method, compared to 1.6, 1.0 and 1.6 for PSGD-J, PSGD-H and AdaGrad, respectively. The Euclidean distance error of FASGD and the proposed FPSGD method is around 5 mm, while about 10 mm for other methods. The p-value of the Wilcoxon signed-rank test of all methods compared to FASGD is smaller than 0.001, indicating a statistically significant difference. The differences are very small for the proposed FPSGD method, i.e. smaller than 0.5 mm, while quite large for PSGD-J, PSGD-H and AdaGrad, i.e. more than 5mm.

The overall results of the experiments on B-spline registration for the SPREAD lung CT data are given in Table VI. For all three resolutions, the proposed method took fewer iterations to obtain the same cost function value as FASGD. Although the proposed method took somewhat longer to estimate the preconditioner compared to FASGD, fewer iterations were required, resulting in an overall improvement of runtime. For the proposed FPSGD method, the overall speed-up is of a factor of 2. The number of iterations used for PSGD-H to obtain the same cost function value as FASGD is lower than both FASGD and the proposed FPSGD method, which can also be observed from the convergence plots in Figure 1c. However, the overhead of computing the preconditioner increased substantially for the PSGD-H method: around 300 seconds for $\sim 10^5$ parameters in resolution 3, while the FPSGD method required $\sim$1s. The speedup factor in terms of overall runtime to obtain the same cost function value as FASGD is consequently much smaller than 1 for PSGD-H. The ED errors in Table VI are evaluated at the end of resolution 3. All methods FASGD, PSGD-H and the proposed FPSGD method obtained a mean ED error around 1.6 mm (within one voxel), while PSGD-J is around 6.9 mm (about 5 voxels). The p-value of the Wilcoxon signed-rank test of PSGD-H, AdaGrad and the proposed FPSGD method compared to FASGD is smaller than 0.05, indicating statistical difference.

TABLE VI

Method Comparison for B-Spline Registration on the SPREAD Lung CT Data. We Used the MSD Dissimilarity Measure, 3 Resolution, 500 Iterations, $\tau = 0.6$ and $\kappa_{\text{MAX}} = 4$

Optimizer	Resolution 1	Resolution 2	Resolution 3	
	Iterations I	Iterations I	Iterations I	ED (mm)
	avg ± std	avg ± std	avg ± std	avg ± std
FASGD	496 ± 0	496 ± 0	493 ± 5	1.66 ± 1.72
PSGD-J	347 ± 183	433 ± 151	496 ± 0	6.86 ± 10.4
PSGD-H	20 ± 7	10 ± 21	61 ± 156	1.48 ± 1.59
AdaGrad	479 ± 54	372 ± 158	237 ± 183	1.49 ± 1.70
FPSGD	232 ± 69	133 ± 32	197 ± 89	1.53 ± 1.54

2) Prostate CT Data: The overall results of the experiments on B-spline registration for the prostate CT data are given in Table VII. Note that PSGD-H is only suitable for the MSD measure which is not used for this dataset. The proposed FPSGD method took fewer iterations to obtain the same cost function value $\mathcal{C}(\widehat{\boldsymbol{\mu}}_{\text{ref}})$ than FASGD and PSGD-J. The speed-up in terms of number of iterations of FPSGD is about 5. The proposed FPSGD method took fewer iterations than AdaGrad in the first two resolutions while FASGD and PSGD-J have almost the same number of iterations. However, the estimation time of the preconditioner for the proposed FPSGD method is about 1.3 seconds longer than AdaGrad, which resulted in a smaller speedup factor than AdaGrad compared to FASGD. The improvements of the proposed FPSGD method compared to other methods in the convergence rate are also shown in Figure 2a.

The Dice overlap of all methods is around 0.87. The p-value of the Wilcoxon signed-rank test of PSGD-J, AdaGrad compared to FASGD is smaller than 0.05, indicating a statistically significant difference. Although significant, the differences are very small for AdaGrad, i.e. less than 0.01, while the performance of PSGD-J is really worse being 9 percent point smaller than FASGD.

3) RIRE Brain Data: Table VIII presents the runtime differences and the mean Euclidean distance error of the RIRE experiments for all methods. We can observe that much fewer iterations are required for the proposed FPSGD method compared to FASGD, while PSGD-J and AdaGrad used almost all number of iterations. The speed-up of the proposed FPSGD method in iterations is a factor of 5. It can also be seen that the

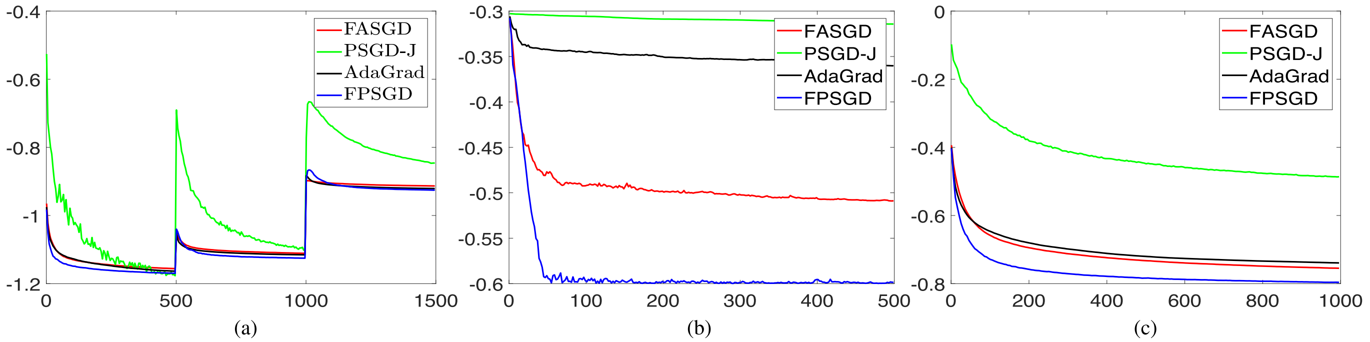


Fig. 2. Convergence plots for different datasets, showing the negative cost function value (MI) against the iteration number. (a) The prostate CT data example. (b) The RIRE brain data example. (c) The BrainWeb simulated data example.

TABLE VII
METHOD COMPARISON FOR B-SPLINE REGISTRATION ON THE PROSTATE CT DATA. WE USED THE MI DISSIMILARITY MEASURE, 3 RESOLUTIONS, 500 ITERATIONS, $\tau = 0.6$ AND $\kappa_{\text{MAX}} = 4$

Optimizer	Resolution 1 Iterations I avg $\pm$ std	Resolution 2 Iterations I avg $\pm$ std	Resolution 3 Iterations I avg $\pm$ std	DSC avg $\pm$ std
FASGD	495 ± 4	496 ± 2	496 ± 0	0.87 ± 0.05
PSGD-J	423 ± 61	492 ± 21	496 ± 0	0.78 ± 0.09
AdaGrad	261 ± 77	214 ± 97	126 ± 43	0.87 ± 0.04
FPSGD	151 ± 27	106 ± 18	132 ± 33	0.87 ± 0.05

speedup in runtime is around 7 for the FPSGD method. The convergence plots in Figure 2b show substantial improvement in convergence rate for the proposed FPSGD method.

The median Euclidean distance of 9 patients before registration is 21.7 mm. PSGD-J and AdaGrad is inferior to the other methods, with a ED around the initial ED, which means that these two methods failed in registration. Although the proposed FPSGD obtained a smaller ED than FASGD, the Wilcoxon signed-rank test shows no statistical difference ($p > 0.05$). PSGD-J and AdaGrad have a significantly larger ED than FASGD.

4) BrainWeb Simulated Brain Data: The results of the BrainWeb experiment are shown in Table IX and Figure 2c. The number of iterations for the proposed FPSGD method to obtain the same cost function value (MI) as FASGD is around 300, resulting in a runtime speed-up of about a factor of 3.3. Both PSGD-J and AdaGrad used about the same number of iterations as FASGD. These improvements can also be observed from the convergence plots in Figure 2c.

The mean residuals of the different methods show a similar result, except for PSGD-J. The Wilcoxon signed-rank test between FASGD and other methods shows a significant statistical difference ($p < 0.05$). However, the difference is very small (around 0.1) for the proposed FPSGD method and AdaGrad, while very large for PSGD-J which means that most registrations of PSGD-J failed.

VII. DISCUSSION

The experimental results show that the proposed FPSGD method works well in both mono-modal as well as multi-modal image registration, in combination with different transformation models and dissimilarity measures, showing that the proposed method is generic for different registration problems. Compared to FASGD which is not preconditioned, the proposed FPSGD method not only obtains the same registration accuracy, but moreover improves the convergence. The performance of PSGD-J is not good as FASGD for affine registration, which might be not a suitable preconditioner for image registration. Without the computational burden of the Hessian matrix calculation and decomposition, the proposed FPSGD method takes much less time than PSGD-H to construct a preconditioner. Additionally, the proposed method requires only a cost function gradient and a set of transformation Jacobians, while PSGD-H also needs the implementation of the self-Hessian. Compared to AdaGrad [20], the proposed method has a slightly longer estimation time but converges faster, resulting in a shorter overall runtime. The proposed method does not need to store and accumulate the squared gradients in the previous step and avoids infinitesimally small updates for later iterations. Most importantly, the proposed FPSGD method is more generic for different modalities and not limited to mono-modal problems like PSGD-H. Note that in this paper the preconditioning methods are discussed in a stochastic setting, however, in principle they can be applied in a deterministic setting also.

Compared to FASGD, the main improvement of the proposed FPSGD method is in the convergence rate, inducing a speedup in runtime of a factor of 2.0-6.0 depending on the application. Specifically, the proposed FPSGD method used 0.25 seconds to obtain the same registration accuracy as FASGD for the affine registration on the SPREAD lung CT data with image sizes of $450 \times 300 \times 130$, while FASGD took 1.45 seconds. For the prostate CT dataset, the proposed method achieved a speedup of around 2 times compared to FASGD for B-spline based registration, resulting in a total runtime of 5.3 s. This enables near real-time daily treatment adaptation, and thus a reduction in treatment margins and robustness criteria that are included in the dosimetric treatment planning. The proposed FPSGD method needs much less computation time for the preconditioner estimation than PSGD-H: ~2 seconds vs ~330 seconds for $\sim 10^5$ transformation parameters. This large difference between different methods in the computation time of preconditioner estimation can be attributed to the

TABLE VIII

METHOD COMPARISON FOR RIGID REGISTRATION ON THE RIRE BRAIN DATASET. WE USED THE MI DISSIMILARITY MEASURE, 1 RESOLUTION, 500 ITERATIONS, $\tau = 0.6$ AND $\kappa_{\text{MAX}} = \infty$

Optimizer	Iterations I	$t_{\text{est}}(s)$	$t_{\text{pure}}(s)$	t_{total} (s)	Speed-up	ED (mm)	p-value
	avg $\pm$ std	avg $\pm$ std	avg $\pm$ std	avg $\pm$ std	avg $\pm$ std	avg $\pm$ std	
FASGD	351 ± 183	0.06 ± 0.01	1.51 ± 0.79	1.57 ± 0.79	-	2.48 ± 2.09	-
PSGD-J	496 ± 0	0.08 ± 0.02	0.98 ± 0.07	1.06 ± 0.08	1.5 ± 0.7	19.4 ± 8.42	0.004
AdaGrad	496 ± 0	0.06 ± 0.01	0.94 ± 0.10	1.00 ± 0.10	1.5 ± 0.7	24.7 ± 13.0	0.008
FPSGD	99 ± 95	0.11 ± 0.02	0.20 ± 0.17	0.31 ± 0.17	7.5 ± 5.5	1.62 ± 0.88	0.301

TABLE IX

METHOD COMPARISON FOR B-SPLINE REGISTRATION ON THE BRAINWEB DATASET. WE USED THE MI DISSIMILARITY MEASURE, 1 RESOLUTION, 1000 ITERATIONS, $\tau = 0.6$ AND $\kappa_{\text{MAX}} = 4$

Optimizer	Iterations I	$t_{\text{est}}(s)$	$t_{\text{pure}}(s)$	$t_{\text{total}}(s)$	Speed-up	Residuals	p-value
	avg $\pm$ std	avg $\pm$ std	avg $\pm$ std	avg $\pm$ std	avg $\pm$ std	avg $\pm$ std	
FASGD	951 ± 34	0.30 ± 0.06	8.30 ± 0.32	8.59 ± 0.33	-	2.42 ± 3.80	-
PSGD-J	996 ± 0	0.48 ± 0.05	7.19 ± 0.05	7.67 ± 0.07	1.1 ± 0.0	8.48 ± 4.91	< 0.001
AdaGrad	989 ± 33	0.32 ± 0.05	7.22 ± 0.32	7.54 ± 0.32	1.1 ± 0.1	2.39 ± 3.80	< 0.001
FPSGD	288 ± 73	0.54 ± 0.04	2.14 ± 0.53	2.68 ± 0.53	3.4 ± 0.9	2.48 ± 3.75	< 0.001

complexity of different methods. For PSGD-H, the complexity is high, mainly due to the Cholesky decomposition of $\mathcal{O}(N_P^3)$, i.e. depending on the number of transformation parameters, while for the FPSGD method the complexity is only linear to the number of parameters $\mathcal{O}(N_P)$. In addition, the runtime per iteration for the PSGD-H method increased to ~780 ms for $N_P \approx 10^5$ transformation parameters, due to the multiplication of a full matrix $\boldsymbol{P}$ instead of only a diagonal matrix for FPSGD (~45 ms per iteration for the MSD dissimilarity measure). We therefore conclude that the proposed FPSGD method converges faster than the FASGD method and is more time-efficient than the PSGD-H method.

The proposed preconditioner shows favorable performance characteristics in terms of runtime, which has tremendous benefits in many applications. This might for example enable real-time daily adaptation of radiation therapy, which has benefits for the patient in terms of adjusted robustness setting and/or reduced treatment margins for the dosimetric treatment planning, potentially resulting in a reduction of adverse side effect of the treatment [2]. In addition, e.g. atlas-based segmentation approaches can gain in performance [1].

There are two parameters that influence the performance of the proposed FPSGD method: the regularization factor τ and the maximum condition number $\kappa_{\max}$. We validated the influence of both parameters experimentally. We showed that the extreme cases ($\tau = 0$ and $\tau = 1$) yielded suboptimal results, indicating that regularization of the preconditioner is required. The proposed regularization method performs a Gaussian smoothing, considering entries with a similar Jacobian response. This choice reflects the observation that transformation parameters that have a similar effect on the displacement, require similar preconditioning, and vice versa. For example, for the affine transformation, rotation and translation require different scaling. The experiments showed that the choice $\tau = 0.6$ yielded good results for all applications. For ill-scaled problems, $\kappa_{\max}$ serves as a safe guard to prevent extreme values in the preconditioner. In the experiment on the SPREAD data, different $\kappa_{\max}$ obtained a similar registration accuracy, however, the convergence has some oscillations for $\kappa_{\max} > 4$ in the second and third resolution in Figure 1a. For the BrainWeb data, best results were acquired with $\kappa_{\max} = 4$ and the convergence plots are also very stable. Overall, the best choice of $\kappa_{\max}$ is between 2 and 4 for nonrigid registration, while $\kappa_{\max} = \infty$ can be used for rigid and affine registration.

To further improve the proposed FPSGD method the following may be considered. Firstly, the proposed preconditioning scheme detailed in Algorithm 2 is very suitable for further acceleration on a Graphics Processing Unit (GPU). It could be easily applied for the parallel computation of the gradient and the preconditioner, therefore this will be beneficial when going to variable preconditioning. Secondly, our method can be combined with the variable preconditioning techniques for difficult problems where the curvature of the cost function changes iteratively, for example just to minimize the expectation of $\tilde{\boldsymbol{g}}_k^T \boldsymbol{P} \tilde{\boldsymbol{g}}_k + \boldsymbol{\mu}_k^T \boldsymbol{P} \boldsymbol{\mu}_k$ recently proposed by Li [16]. Furthermore, a stopping condition other than the number of iterations will be required to practically take advantage of the convergence improvements. An interesting option suitable in a stochastic setting is a moving average of the noisy gradients over a few iterations.

VIII. CONCLUSION

In this paper, we proposed a generic preconditioner estimation method for the stochastic gradient descent optimizers used in medical image registration. Based on the observed distribution of the voxel displacements, this method automatically constructs a diagonal preconditioner, avoiding the time-consuming calculation of the Hessian matrix. All tested methods obtained a similar final registration accuracy in all tested datasets. The proposed FPSGD optimizer, however, outperforms FASGD and PSGD-J in terms of convergence rate, while yielding a similar computational overhead. While a previous method (PSGD-H) even further reduces the required number of iterations, it comes at a substantial overhead in computing the preconditioner, especially for high dimensional transformations. Additionally, PSGD-H can only be used in mono-modal problems and requires the implementation of a Hessian matrix computation.

We conclude that the proposed method can act as a generic preconditioner for optimization in registration methods, yielding similar accuracy as gradient descent routines while substantially improving the convergence rate.

Acknowledgments

The RIRE project is acknowledged for providing a platform for rigid registration evaluation. Dr. M.E. Bakker and J. Stolk are acknowledged for providing a ground truth for the SPREAD study data used in this paper.

References

[1] M. A. Viergever, A. Maintz, S. Klein, K. Murphy, M. Staring, and J. P. W. Pluim, "A survey of medical image registration—Under review," *Med. Image Anal.*, vol. 33, pp. 140–144, Oct. 2016.

[2] W. Li, D. A. Jaffray, G. Wilson, and D. Moseley, "How long does it take? An analysis of volumetric image assessment time," *Radiotherapy Oncol.*, vol. 119, no. 1, pp. 150–153, 2016.

[3] S. Thörnqvist *et al.*, "Degradation of target coverage due to inter-fraction motion during intensity-modulated proton therapy of prostate and elective targets," *Acta Oncol.*, vol. 52, no. 3, pp. 521–527, 2013.

[4] P. Kupelian *et al.*, "Multi-institutional clinical experience with the calypso system in localization and continuous, real-time monitoring of the prostate gland during external radiotherapy," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 67, no. 4, pp. 1088–1098, 2007.

[5] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006, pp. 497–528.

[6] 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.

[7] V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 32–43, Sep. 2014.

[8] S. Klein, J. P. W. Pluim, M. Staring, and M. A. Viergever, "Adaptive stochastic gradient descent optimisation for image registration," *Int. J. Comput. Vis.*, vol. 81, no. 3, pp. 227–239, 2009.

[9] M. Benzi, "Preconditioning techniques for large linear systems: A survey," *J. Comput. Phys.*, vol. 182, no. 2, pp. 418–477, 2002.

[10] C. Li, C. Chen, D. Carlson, and L. Carin, "Preconditioned stochastic gradient langevin dynamics for deep neural networks," in *Proc. 30th Conf. Artif. Intell. (AAAI)*, 2015, pp. 1788–1794.

[11] H. Jiang, G. Huang, P. A. Wilford, and L. Yu, "Constrained and preconditioned stochastic gradient method," *IEEE Trans. Signal Process.*, vol. 63, no. 10, pp. 2678–2691, May 2015.

[12] D. E. Carlson, E. Collins, Y.-P. Hsieh, L. Carin, and V. Cevher, "Preconditioned spectral descent for deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2971–2979.

[13] Y. Dauphin, H. De Vries, and Y. Bengio, "Equilibrated adaptive learning rates for non-convex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1504–1512.

[14] D. Zikic, M. Baust, A. Kamen, and N. Navab, "A general preconditioning scheme for difference measures in deformable registration," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 49–56.

[15] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, "A stochastic quasi-Newton method for large-scale optimization," *SIAM J. Optim.*, vol. 26, no. 2, pp. 1008–1031, 2016.

[16] X. Li, "Preconditioned stochastic gradient descent," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1454–1466, May 2018.

[17] Y. Qiao, Z. Sun, B. P. Lelieveldt, and M. Staring, "A stochastic quasi-Newton method for non-rigid image registration," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*, 2015, pp. 297–304.

[18] S. Klein, M. Staring, P. Andersson, and J. P. Pluim, "Preconditioned stochastic gradient descent optimisation for monomodal image registration," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2011, pp. 549–556.

[19] Y. Qiao, B. van Lew, B. P. F. Lelieveldt, and M. Staring, "Fast automatic step size estimation for gradient descent optimization of image registration," *IEEE Trans. Med. Imag.*, vol. 35, no. 2, pp. 391–403, Feb. 2016.

[20] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.

[21] D. Vysochanskij and Y. I. Petunin, "Justification of the 3σ rule for unimodal distributions," *Theory Probab. Math. Statist.*, vol. 21, pp. 25–36, 1980. [Online]. Available: http://probability.univ.kiev.ua/tims/ and https://www.statindex.org/journals/1862/21.bib?action=volume&controller=journals&id=1862&vol=21

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

[23] 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.

[24] 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.

[25] C. A. Cocosco, V. Kollokian, R. K.-S. Kwan, and A. C. Evans, "BrainWeb: Online interface to a 3D MRI simulated brain database," *NeuroImage*, vol. 5, no. 4, p. S425, 1997.

[26] S. M. Smith, "Fast robust automated brain extraction," *Hum. Brain Mapping*, vol. 17, no. 3, pp. 143–155, Sep. 2002.

[27] 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.