Distribution Agreement

Huiying Zhu December 1, 2017

Application of Global Optimization to Image Registration

By

Huiying Zhu

Lars Ruthotto

Advisor

Department of Mathematics and Computer Science

Lars Ruthotto

Advisor

Manuela Manetta

Co-Advisor

Dwight Duffus

Committee Member

Marianne K. Lancaster

Committee Member

2017

Application of Global Optimization to Image Registration

By

Huiying Zhu

Lars Ruthotto

Advisor

An Abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelors of Science with Honors

Department of Mathematics and Computer Science

2017

Abstract

Application of Global Optimization to Image Registration

By Huiying Zhu

Given two images, image registration aims to transform an image into a given reference image so that the two images look alike. This technique is vital in many applications, such as medical imaging and astronomy. Finding the best transformation can be phrased as solving a mathematical optimization problem. Due to the non-convexity of the objective function, commonly employed optimization techniques often generate local minimizers, limiting the accuracy of the registration. This thesis evaluates the applicability of a global optimization method, called DDNCID, for image registration. Direct application of DDNCID in image registration could cause minimizers to be infeasible. Thus, a focus of this thesis is to add a bound constraint by imposing a barrier function into the objective function to extend DDNCID.

Application of Global Optimization to Image Registration

By

Huiying Zhu

Lars Ruthotto

Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelors of Science with Honors

Department of Mathematics and Computer Science

2017

Contents

Chapter 1

Introduction

Given two images, image registration aims to transform one of the two images into the other image so that the two images look alike [1]. Image registration has a large number of applications, ranging across astronomy, biology, chemistry, criminology or any area involving imaging techniques [2]. In medical imaging, registration is widely used to align images to compare data from different modalities such as computer tomography (CT), magnetic resonance imaging (MRI) and positron emission tomography (PET), before data can be compared or combined [3].

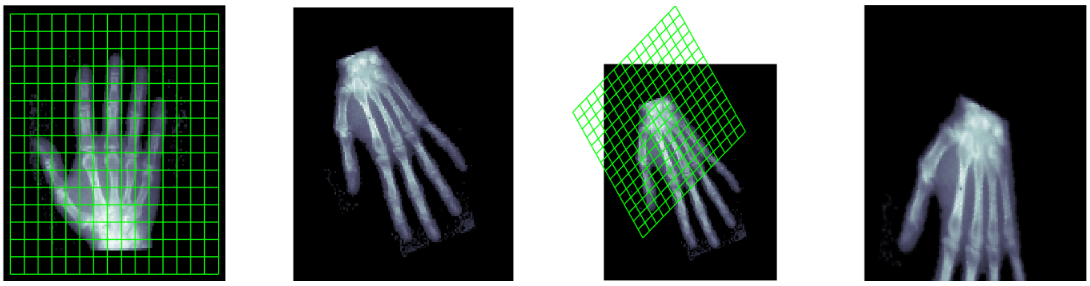
Many fields view image registration as a useful tool. In particular, it could be applied in radiology to foster processing as we continue to improve and rely on imaging techniques. Significant emphasis has been put into research activity to align images from the same or different subjects to correct scanner-induced geometric distortion [4]. Though required accuracy varies between applications, it is good to ensure acceptable accuracy. Thus, we are looking for a registration technique that provides good indication of errors and requires minimal manual input.

When interpreted mathematically, image registration is phrased as a non-convex optimization and ill-posed problem [2, 5]. The notion of well-posedness is defined as the problem has a solution, the solution is unique and depends continuously on the data [5]. Therefore, existence of multiple local minima is common in such problems. To determine if a solution is the global minimum without visual assessment is challenging in image registration, as various image registration models are available and we know little about expected accuracy [4]. Due to the vast application of image registration and the variety of registration purposes, such as intensity based and geometric features, no general theory has been established [1]. The foci of a particular applications can be different and can range from computing time, image features, memory to accuracy of a model [4].

1.1 Motivation

The motivation for this thesis is to explore the global optimal transformation of a given image, called a template image, that is most similar to the given reference image. As different transformation techniques are developed for different purposes, to establish a standard method for image registration can be difficult. Furthermore, registration problem can be complex due to the existence of multiple local minima. Even though attempts have been made to reduce the chance of being trapped by local minima such as omitting image details [2], current optimization techniques applied to image registration are still not perfect in solving optimization problems. In this thesis, the goal is to apply global optimization to improve the accuracy of image registration.

Particularly, this thesis aims to extend the current method to avoid finding a local minimum instead of a global solution. To explore the best transformation, we implement and extend the method named Double Descent and Color Intermittent Diffusion (DDNCID) [6]. We aim to modify the process of the DDNCID [6] in order to apply it to image registration.



(a) Reference Image and regular grid (green)
(b) Template Image
(c) Template Image and Transformed Grid (green)
(d) Transformed Template

Figure 1.1: A 2D Hand Example solved by local optimization (Gauss-Newton) [1]

For example, in Figure 1.1, we are given a reference image and a template image. The solution found by the Gauss-Newton method as a numerical optimizer is only a local minimizer [1]. This is a typical example that illustrates the limitation of some numerical solvers dependent on initial guesses.

1.2 Related Work

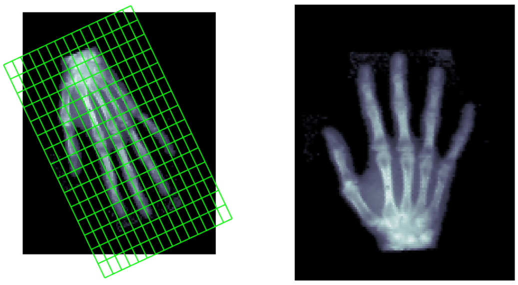
Due to the vast applications of image registration and different features of template and reference images, transformation models have been discussed to determine correspondence between point features [3]. Examples of transformation functions are rigid transformation, affine transformation, thin-plate spline (TPS) transformation and multiquadric transformation [1, 3].

As image registration is interpreted mathematically as an optimization problem, the objective function often involves a cost function $D(w)$ to measure similarity, a regularization $S(y)$ and penalty functions $P(w)$ to avoid undesirable transformations [2]: $J(y) = D(y) + S(y) + P(y)$, subject to $y \in M$, where M is a set of transformations.

It is true that Point-Landmark registration has an understood error propagation [4], but it also requires more effort to manually align points of the reference image with the points of the corresponding template image. For approaches other than landmark registration, less information is given about their of accuracy [4].

In Flexible Algorithm for Image Registration (FAIR) [1], we often use a Gauss-Newton scheme as numerical optimization method in parameterized transformation. Although practical applications of regularizers have been used to make the registration well-posed [5], results of optimizing the objective function will depend on the starting point, the algorithm and implementation [1].

When put into practice, the Gauss-Newton scheme with line search [1] will find a minimum dependent on a starting guess. As image registration is a non-convex problem, the Gauss-Newton scheme can only provide a local search in the descending direction of the starting guess. In Figure 1.1, given a reference image and a template image, the Gauss-Newton method only yields a local but not a global minimum when minimizing the objective function.



(a) Reference Image and regular grid (green)

(b) Template Image

(c) Template Image and Transformed Grid (green)

(d) Transformed Template

Figure 1.2: Affine Registration Problem solved by Global Optimization

1.3 Contribution

In this thesis, we test an available global optimization method on image registration. A global optimization should theoretically be able to find global minima regardless of the starting guess. We implement the DDNCID method for global optimization on image registration [6].

As the current version of DDNCID does not cover bounded problems, we control the feasibility of solutions by adding a barrier function into the objective function.

As we test on different starting guesses on Rigid and Affine transformations, with the global solution already known to us, DDNCID is a good approach to find the global solutions. Solving exactly the same problem as in 1.1, DDNCID with log barrier constraint is able to find the global minimizer that maps our template image back to the reference image.

When compared to other optimization methods such as the Gauss-Newton scheme, the advantage of DDNCID with log barrier constraint is that it does not rely on the starting guess. Even starting with a distorted guess, our global optimization method is able to find the optimal transformation. The goal of global optimization is to increase the accuracy of the optimization process and guarantee the solution to be global and feasible.

Further considerations such as optimizing the algorithms and imposing direct calculation of the Hessian are expected to be done to increase the efficiency and accuracy of current work.

1.4 Outline

In Chapter 2, we discuss the current method of optimization including global search and local optimization built in FAIR [1]. Then in Chapter 3, we extend the current method of DDNCID with the addition of a log barrier constraint. Several numerical experiments based on different image transformation models are shown in Chapter 4. Chapter 5 includes a conclusion and future directions.

Chapter 2

Optimization in Image Registration

2.1 Image Registration

As mentioned in Chapter 1, image registration is a spatial transformation from a template image to a reference image so that two images look alike. A successful transformation guarantees the relationship between the position of a feature in one image or coordinate space and the position of the corresponding feature in another image or coordinate space [4]. Additionally, transformation helps to compare data such as intensity and average in a region of interest.

Geometrically, a transformation is defined as a mapping of corresponding points from one image to another image. As multiple imaging modalities exist and geometrical distortions of objects vary, various transformation models are constructed for convenience and different applications. The transformation problem can be solved by optimizing the cost function (objective function), which sums the similarity of the transformed template image and reference image, and regularization, which prevents undesirable and extreme transformations [2].

2.1.1 Grids

For data given on a spatial dimension d, we define a spatial domain $\Omega \subset \mathbb{R}^d$ as the region of the coordinate system where the template image is placed and a grid as a partitioning of the domain into a number of congruent cells [1]. For a data set of size n, in the following, $\boldsymbol{x} = [x_1^1; \cdots ; x_n^1; \cdots ; x_1^d; \cdots ; x_n^d] \in \mathbb{R}^{nd}$, is a vector representing the coordinates of the centers of the cells $x_j = [x_j^1, \cdots , x_j^d]$, for $j = 1, \cdots , n$. In parameterized image registration, we need to evaluate the value of each transformed point $y(w)$ in order to calculate the similarity of the transformed template image and the reference image.

2.1.2 Template and Reference

Using a template image and a reference image, we define $\texttt{dataT} \in \mathbb{R}^n$ and $\texttt{dataR} \in \mathbb{R}^n$ as discrete data on a regular grid, where n is the number of centered cells. Then $\mathcal{T}$ is defined as the mapping of template image and $\mathcal{R}$ as the mapping of the reference image from domain $\Omega \subset \mathbb{R}^d$ into $\mathbb{R}$. For any point $x_j \in \Omega \subset \mathbb{R}^d$ to the set of real numbers $\mathbb{R}$, $\mathcal{T}$ and $\mathcal{R}$ are interpolation functions that satisfy,

$$\mathcal{T}(x_j) = \texttt{dataT}(j); \tag{2.1}$$

$$\mathcal{R}(x_j) = \texttt{dataR}(j), \tag{2.2}$$

for $j = 1, 2, ..., n$, where n is the number of cells.

For example, 2D images have spatial dimension $d = 2$. For any point on the grid, the mapped value represents the corresponding grey scale value of that point. The purpose of the interpolation function is to transform the data from discrete points into a continuous function and approximate the data on undefined points.

In our thesis, we use cubic B-spline interpolation to ensure the grid points are differentiable. Thus, for 2D data with discretization size $m = [m^1\, m^2]$, the interpolation function can be expanded as [1]:

$$\mathcal{T}(x) = \mathcal{T}^{Spline}(x) = \sum_{j_2=1}^{m^2} \sum_{j_1=1}^{m^1} c_{j_1,j_2}\, b^{j^1}(x^1) b^{j^2}(x^2) \tag{2.3}$$

$$b\,(x) = \begin{cases} (x+2)^3 & -2 \leq\ x\ <\ -1 \\ -x^3 - 2(x+1)^3 + 6(x+1) & -1 \leq\ x\ <\ -1 \\ x^3 + 2(x-1)^3 - 6(x-1) & 0 \leq\ x\ <\ -1 \\ (2-x)^3 & 1 \leq\ x\ <\ 2 \\ 0 & \text{else} \end{cases}$$

2.1.3 Rigid and Affine Transformations

In this thesis, our experiments are based on parameterized image registration (PIR) based on FAIR [1]. Though numerous transformation models are available, this thesis only applies rigid transformation and affine transformation. Other transformation models such as the spline-based transformation in higher dimensions and addition of regularization are expected to be tested for future work. These transformation models are also important since rigid registration is in general very limited and generally the search space might be explicitly given [2].

Rigid transformation is a parameterized transformation that only allows shifting and rotations. For $d = 2$, parameter $w = [w_1; w_2; w_3] \in \mathbb{R}^3$ and each center of the grids $x = [x^1; x^2] \in \mathbb{R}^2$, let us define the transformation $y(w) \in \mathbb{R}^2$ as follows

$$y^1(w) = \cos(w_1)x^1 - \sin(w_1)x^2 + w_2 \tag{2.4}$$

$$y^2(w) = \cos(w_1)x^1 + \sin(w_1)x^2 + w_3. \tag{2.5}$$

(a) Reference Image (b) Template Image (c) Template with transformed grid

Figure 2.1: 2D rigid transformation example: data from a histological serial section provided in FAIR [1]

Affine transformation is a transformation more flexible than rigid transformation. It allows rotation, shearing and individual scaling [1]. An affine linear transformation $y(w) \in \mathbb{R}^2$ is defined as

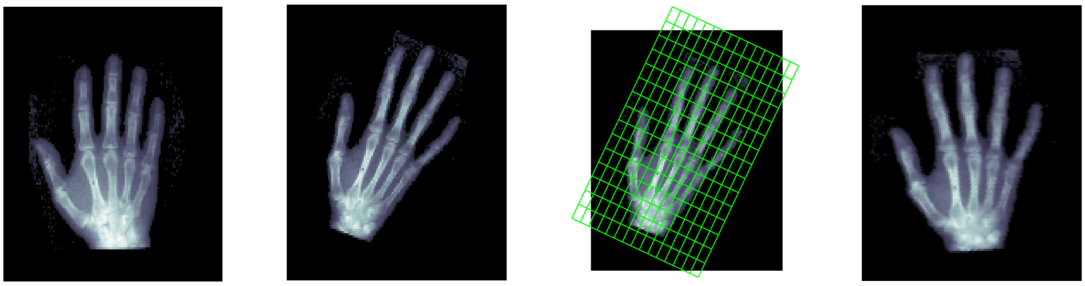
$$y(w) = \begin{bmatrix} x^1 & x^2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x^1 & x^2 & 1 \end{bmatrix} w, \tag{2.6}$$

where $w = [w_1; w_2; \cdots; w_6] \in \mathbb{R}^6$.

2.1.4 Objective Function

The objective function J to be optimized is defined as

$$J(w) = D(w) + S(w),$$



(a) Reference Image (b) Template Image (c) Transformed Template with grid (d) Transformed Template with grid

Figure 2.2: 2D affine transformation example: 2D hand data provided in [1]

where $D(w)$ measures the difference between each corresponding point in the transformed template image and reference image and $S(w)$ is the regularizer that penalizes undesirable transformation. In this thesis, we use Sum of Squared Difference (SSD) as the $D(w)$ [1]. SSD is defined as the sum of absolute differences on all the points on the transformed grid:

$$D(w) \quad = \frac{1}{2}\int_{\Omega}(\mathcal{T}\left(y\left(w,x\right)\right) - \mathcal{R}\left(x\right))^2\, dx.$$

Instead of calculating the integral analytically, we use a midpoint quadrature rule to numerically approximate the integral [1]:

$$D(w) = \frac{1}{2}\int_{\Omega}(\mathcal{T}\left(y\left(w,x\right)\right) - \mathcal{R}\left(x\right))^2\, dx \approx \frac{1}{2}\, h\, \|(T(\boldsymbol{y}(w,\boldsymbol{x})) - R(\boldsymbol{x}))\|^2,$$

with $h = h_1\, h_2 \cdots h_d$, $T(\boldsymbol{y}) = [\mathcal{T}\left(y_j\right)]_{j=1}^{n} = [\mathcal{T}\left(y(w,x_j)\right)]_{j=1}^{n}$ and $R(\boldsymbol{x}) = [\mathcal{R}(x_j)]_{j=1}^{n}$, where $\boldsymbol{y}(w,\boldsymbol{x})$ is the transformed grid from the regular grid $\boldsymbol{x}$ and h_1, $h_2, \cdots, h_n$ are the widths of the cell centered grid. As we already discussed in the Chapter 1, image registration has many solutions. Thus, we could add some additional conditions to rule out undesirable solutions by introducing a regularizer [1]. To simplify our registration problem, we set the regularizer $S(w) = 0$.

Therefore, our optimization problem becomes

$$\min_{w} \quad J(w) = \frac{1}{2}\, h\, \|(T(\boldsymbol{y}(w,\boldsymbol{x})) - R(\boldsymbol{x}))\|^2.$$

FAIR [1] provides optimization methods such as the Gauss-Newton method and the Steepest Descent (Gradient Descent) method as numerical optimization schemes to minimize objective function and a multi-level strategy to reduce computation time and omit some cases of local minima. However, the success of the Gauss-Newton scheme is dependent on the initial starting guess. In FAIR, Gauss-Newton is achieved by transforming our distance measure (SSD) into the following [1]:

$$J(w) = \psi(r(w)),$$

where $r = T(y\,(w, \boldsymbol{x}) - R(\boldsymbol{x})$ and $\psi = \frac{1}{2}\, r^T\, r$. It optimizes a quadratic function

$$\widehat{J}(w + dw) = J + dJ\, dw + \frac{1}{2}\, dw^T\, H\, dw \;\approx\; J(w + dw),$$

where the approximate Hessian is $H = dr^T d^2\psi\, dr$. The descent direction is found by solving the linear system [7]

$$H\, dw = -\nabla J.$$

When dw is small, we have a good approximation of our original function. However, when dw is large, the quadratic function $\widehat{J}$ cannot guarantee a good approximation of our original problem.

(a) Reference Image (b) Template Image (c) Transformed Template Image

Figure 2.3: Given template image and reference image, Gauss-Newton with initial guess w=0 finds a solution that is a local minimum.

Here, we offer the simplest example in a 1D rotation transformation model so that we are able to plot the objective function and determine the global minimum. If the starting guess is close to a local but not global minimum, Gauss-Newton is likely to fail. In Figure 2.3, from the starting guess $w = 0$, Gauss-Newton would return a local minimum of the objective function in Figure 2.4.

2.2 Double Descent and Color Intermittent Diffusion

In this thesis, we apply a global optimization method called Double Descent and Color Intermittent Diffusion (DDNCID) [6], developed by L.Deici, M. Manetta and H.Zhou, to the image registration problem. As Figure 2.3 and Figure 2.4 show, the success of Gauss-Newton is dependent on the choice of the initial guess. Thus we take the uncertainty of Gauss-Newton scheme as a motivation to explore a global optimization technique in our problem.

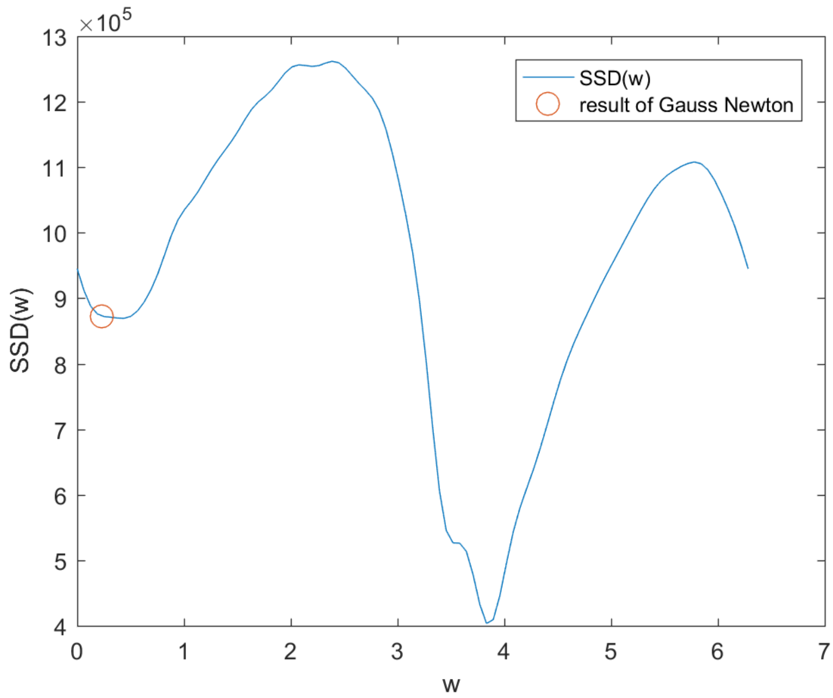


Figure 2.4: graph of SSD and result of Gauss-Newton with initial guessing w=0

2.2.1 Mathematical Background of DDNCID

Given the objective function $J(w) : \mathbb{R}^p \to \mathbb{R}$, where p is the dimension of our transformation parameters, DDNCID [6] divides exploration of minima into two steps: local search and escape from basins of attraction. Let $G(w) = \frac{1}{2}(\nabla J(w))^T(\nabla J(w))$ be an auxiliary function. Before we discuss DDNCID in detail, we provide some important concepts.

A point w^* is called a ***local minimum*** of J over domain U if there exists $r > 0$ for which $J(w^*) \leq J(w)$ for any $w \in U \cap B(w^*, r)$. If $J(w^*) < J(w)$ for any $w* \neq w \in U \cap B(x^*, r)$, w^* is a ***strict*** local minimum [8].

A point w^* is called a ***local maximum*** of J over domain U if there exists $r > 0$ for which $J(w^*) \geq J(w)$ for any $w \in U \cap B(w^*, r)$. If $J(w^*) > J(w)$ for any $w^* \neq w \in U \cap B(w^*, r)$, w^* is a ***strict*** local maximum [8].

In optimization, a point w^* in the domain Ω of a function $J(w)$, which is differentiable over some neighborhood of w^*, is a ***critical point*** of $J(w)$ if $\nabla J(w^*) = 0$ [8]. Here, the condition is equivalent to $G(w^*) = 0$.

After finding a critical point w^*, we could use sufficient second order optimality conditions to classify whether the point is a maximum or a minimum.

1. If $\nabla^2 J(w^*) > 0$, then w^* is a strict local minimum of J over Ω.

2. If $\nabla^2 J(w^*) < 0$, then w^* is a strict local maximum of J over Ω.

We identify each critical point by finding the largest λ_{big} and smallest eigenvalue λ_{small} of the Hessian matrix $\nabla^2 J$. If both λ_{big} and λ_{small} are positive, which means the matrix is positive definite, the critical point is a minimizer. If the signs of λ_{big} and λ_{small} are different, we categorize the point to be a saddle point. If both λ_{big} and λ_{small} are negative, which means the Hessian is negative definite, the critical point is considered as a maximizer [6, 8].

The escape process consists of perturbing points in the neighborhood of the critical point found to escape from "its basins of attraction" [6]. When the new iterant has a different signature of the Hessian compared to the Hessian of the last initial point, we say the new iterant escapes from the basin of attraction of that critical point.

1. With a random initial guess w_0, the optimization starts by searching for a minimum using double descent by decreasing both J and G. Then the new point is stored in an array $Wcri$.

2. A random point is selected from $Wcri$. According to the identification of this point (maximizer, minimizer or saddle point), the algorithm adds colored diffusion to the critical point in order to exit its basin of attraction. Then a local search begins for the next critical point, which is then stored in the array of critical points mentioned above.

3. Step 2 is repeated by randomly selecting another critical point in the array $Wcri$. We continue the basin escaping and local search until the number of iterations reaches the maximum iteration allowance.

2.2.2 Local Search and Basins Escape

During local search of minimizers, DDNCID uses double descent to decrease both J and G. At the beginning of each iteration, a random critical point w_0 is selected from $Wcri$ and the algorithm performs the following method to get to another critical point depending on whether the critical point is a minimizer, maximizer or saddle point:

1. If $w_0 \in \mathbb{R}^n$ is a minimum:

   (a) We find the eigenvalues of the Hessian $H(w_0)$ in descending order: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. First, we perturb w_0:
   $$w_1 = w_0 + \alpha\,\sigma\sqrt{h}\,W, \tag{2.7}$$
   where α is a scalar, $W \in N(0,1)^p$ is a vector with normally distributed random numbers, $\sigma = v_1 v_1^T$ with v_1 the eigenvector associated with λ_1 and h is a step size

which is 1 by default. Here, we choose the eigenvector related to the largest positive eigenvalue of the Hessian so that the diffusion step (escape) can be as quick as possible.

(b) Using QR factorization to find $H^{\dagger}(w_k)\nabla g(w_k)$, where $H^{\dagger}(w_k)$ is the pseudo-inverse of the Hessian $H(w_k)$, we find the step size h with $|G(x_k - h(H^{\dagger}(w_k))| < |G(x_k)|$ and update w_{k+1}:

$$w_{k+1} = w_k - h(H^{\dagger}(w_k)\,\nabla g(w_k)) + \alpha\,\sigma\sqrt{h}\,W. \tag{2.8}$$

The choice of the pseudoinverse is dictated by the possibility of having a singular Hessian.

(c) We iterate the previous step until we find some w^* such that $H(w^*)$ is no longer symmetric positive definite. This means that we escape from the basin of attraction of the local minimum. Then we use (damped) Newton with the aim of reaching a saddle point. In this case, h and v are chosen by only imposing a decreasing condition on G.

(d) We store the new critical point in the array. Then we randomly select a critical point and iterate 1.(a), 1.(b) and 1.(c), if the critical point is a minimum.

2. If $w_0 \in \mathbb{R}^n$ is a saddle point:

(a) The goal is to get out of the neighborhood of the saddle point. As for a local minimizer, we first perturb w_0:

$$w_1 = w_0 + \alpha\,\sigma\sqrt{h}\,W.$$

where α is a scalar, $W \in N(0,1)^p$ is a random vector, $\sigma = v_n v_n^T$ and v_n is the eigenvector associated with λ_n. The purpose of choosing the eigenvector related to the smallest eigenvalue is to move away from the saddle point as quickly as possible and decrease J.

(b) The second step consists of perturbing w_1 by using means of colored diffusion and a descent direction (either double descent ($v = H^{\dagger}(w_k)\nabla g(w_k)$) or gradient descent($v = -\nabla g(w_k)$)). The diffusion step is repeated as follows until the Hessian is symmetric positive definite.

i. Double Descent: Find step size h such that $|G(\hat{w}_{k+1})| < |G(w_k)|$ and $|g(\hat{w}_{k+1})| < |g(w_k)|$, where $\hat{w}_{k+1} = w_k + hv$ and update

$$w_{k+1} = w_k + hv + \alpha\,\sigma\sqrt{h}\,W.$$

ii. Gradient Descent: Find step size h such that $|g(\hat{w}_{k+1})| < |g(w_k)|$ and update

$$w_{k+1} = w_k + hv + \alpha\,\sigma\sqrt{h}\,W.$$

Once the point leaves the basins of attraction or reaches the maximum number of iterations, DDNCID uses Double Descent again to find direction v and step size h to decrease both g and G. Then we store the final result in the array and continue with another randomly selected critical point.

3. If $w_0 \in \mathbb{R}^n$ is a maximum:

   (a) We find the eigenvalues of the Hessian $H(w_0)$ in descending order: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. First, we perturb w_0 so that

   $$w_1 = w_0 + \alpha\,\sigma\sqrt{h}\,W,$$

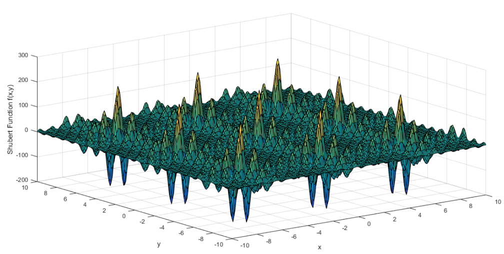
   where α is a scalar, $W \in N(0,1)^p$ is a random vector, $\sigma = v_n v_n^T$ and v_n is the eigenvector associated with λ_n. Here, we choose the eigenvector related to the smallest eigenvalue (negative) so that the diffusion step can be as quick as possible.

   (b) Evaluating $H^{\dagger}(w_k)\nabla g(w_k)$, we find h with $|G(\hat{x}_{k+1})| < |G(x_k)|$ and update w_{k+1}

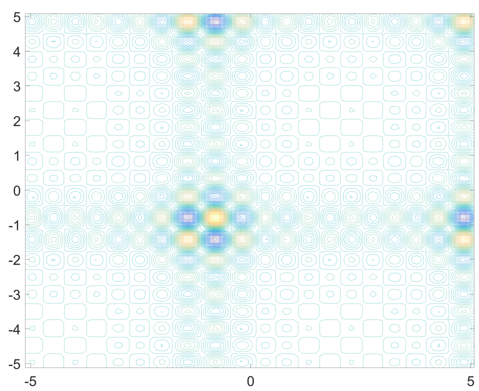
   $$w_{k+1} = w_k - h(H^{\dagger}(w_k)\nabla g(w_k)) + \alpha\,\sigma\sqrt{h}\,W$$

   (c) We continue the previous step until for some w^*, $H(w^*)$ is not symmetric positive definite. This means that we escape from the basin of attraction of the local minimum. Then we assure the decrease of G, so that $G(w_k + hv) < G(x)$, using Newton's direction v.

   (d) We store the new critical point in the array $Wcri$ and continue the new iteration by randomly selecting a critical point in it.



(a) Graph of the Shubert function



(b) Contour of the Shubert function

Figure 2.5: Shubert Function

Here, we test the algorithm with the Shubert function [6], which is given by:

$$g(x,y) = \Big(\sum_{i=1}^{5} i\, cos\,[(i+1)\,x + i\,] \Big) \Big(\sum_{i=1}^{5} i\, cos\,[(i+1)\,y + i\,] \Big). \tag{2.9}$$

From Figure 2.5, in the domain $[-6, 6] \times [-6, 6]$ this function contains multiple global minima and critical points. By starting at a random point and setting the number of iterations to be 500, DDNCID finds at least a global minimum in the array of minimizers.

As we can see in Figure 2.5, the Shubert function not only has many critical points (minima, maxima and saddle points), but also it shows multiple global minima. DDNCID is able to explore the landscape of the function and find the global minima, in particular.

2.2.3 Challenge of Implementation of DDNCID in Image Registration

During the initial implementation, DDNCID provides many critical points which shift the template far away from the domain Ω. This is because DDNCID, as presented in [6], does not consider constrained optimization. Though image registration is not initially defined as a constraint optimization problem, extreme shifting or rotation away from the image frame is considered now as an infeasible solution. In our implementation, DDNCID can generate undesirable solutions as it gradually moves the iterant away from the domain. Therefore, we set a boundary for each image registration model and propose a method to avoid the image being moved away from the boundary.

In terms of the Hessian, DDNCID uses finite difference to approximate the matrix H. If the rate of change is huge regarding to the default difference, the approximation could potentially lead the classification of the Hessian to be misleading. As the iteration and termination of the escape process depend on the identity of the Hessian, it is essential for us to have a good approximation of it.

The cost of local search using double descent is expensive and we may consider other local optimization methods that are more efficient than double descent. We replace double descent by Gauss-Newton for the local search and will discuss this in detail in the next chapter.

Chapter 3

Global Optimization with Log Barrier Constraint

In Chapter 2, we see that in some transformation models, DDNCID [6] can possibly generate undesired shifting of the template image away from the defined domain Ω. To avoid the transformation being shifted away from the boundary, we set some constraints w_{up} and/or w_{low} to control the parameters w_i that are related to shifting.

3.1 Logarithmic Barrier Method

The current method of DDNCID is not designed for constrained optimization problems. Numerical experiments show that unconstrained global optimization can possibly update the transformation to be infeasible (away from Ω). Therefore, we are seeking an alternative, simple approach to translate a constrained optimization problem into a non-constrained problem. Since DDNCID [6] relies heavily on the gradient and the Hessian, a good approximation of these terms is crucial to the success of application of DDNCID [6] to image registration problems.

The idea of a log barrier constraint is similar to regularization. We add a term $P(w)$, which allows us to add information of constraints into the objective function $J(w) : \mathbb{R}^p \to \mathbb{R}$, where p is the number of unknowns in our transformation parameter w.

3.1.1 Penalty Function

In our image registration problems, we pick the constraints w_{low} as the lower bound of w and w_{up} as the upper bound of w. The choice of boundary is determined by the image model and the domain Ω of the function. The idea is to control the entries that affect shifting of the template image. With a "barrier parameter" α [7], the barrier function is generally defined by

$$P(w, \alpha) = \alpha(-\log(w - w_{low}) - \log(w_{up} - w)) \tag{3.1}$$

$$= \alpha(-\sum_{i=1}^{n} \log(w_i - (w_{low})_i) - \sum_{j=1}^{n} . \log((w_{up})_j - w_j)) \tag{3.2}$$

The logarithmic barrier functions have the following properties [7]:

1. $P(w)$ is differentiable inside the boundary;
2. $P(w) \to \infty$ as $w \to w_{low}$ or $w \to w_{up}$.

In MATLAB, we assign infinity to $P(w_k)$ when $\forall$ indices $i \leq n, \exists\, j$ such that $(w_k)_j \leq (w_{low})_j$ or $w_j \geq (w_{up})_j$.

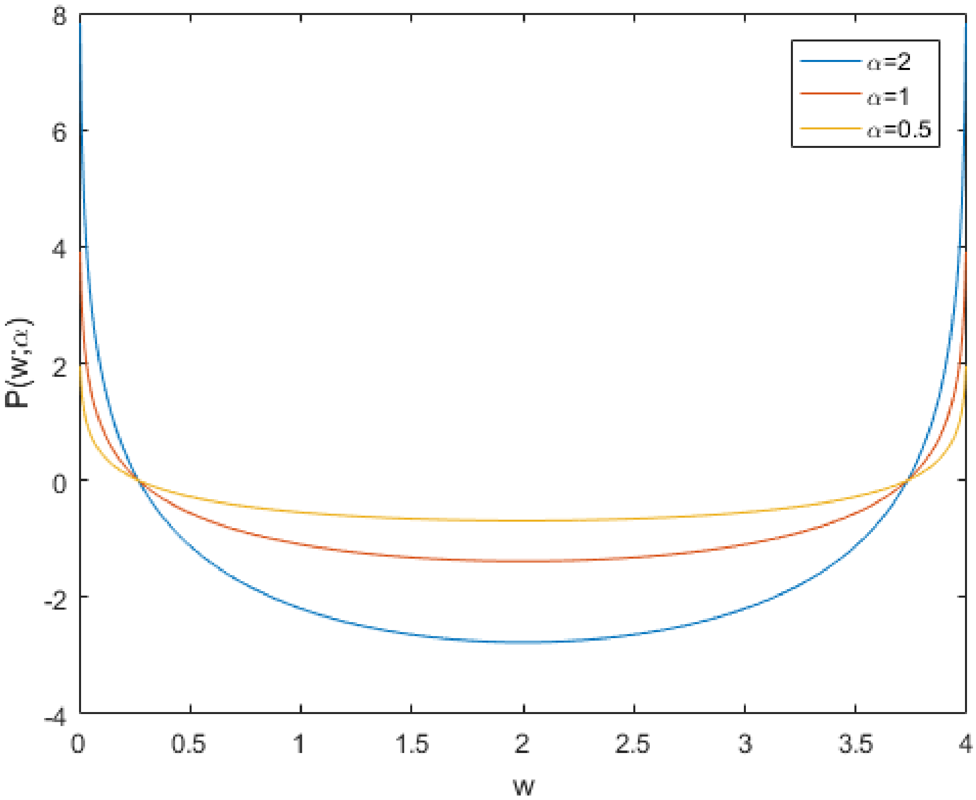


Figure 3.1: plot of $P(w;\alpha)$ in 1D with $w_{low} = 0$ and $w_{up} = 4$ with $\alpha = 2, 1, 0.5$

Adding the logarithmic barrier function into the objective function $J(w)$, our new objective function becomes

$$J(w;\alpha) = D(w) + P(w;\alpha). \tag{3.3}$$

In Figure 3.1, we have a plot of the function with $w_{low} = 0$ and $w_{up} = 4$:

$$S(w;\alpha) = \alpha\Big(-\log(w - w_{low}) - \log(w_{up} - w)\Big).$$

Assigning a sequence of α's, the function becomes flatter when α decreases. When the log barrier constraint is added to the objective function, $J(w)$, $P(w)$ will increase dramatically as w approaches the boundary. Moreover, when α is small, $P(w)$ does not have significant effect on $J(w)$ for w that is not around the boundary.

Thus, under certain conditions, the minimizer of $J(w;\alpha)$, which is denoted by w_u*, approaches a solution of our original problem $\min\limits_{w} J(w)$.

3.1.2 Gradient and Hessian of $J(w;\alpha)$

Now, we have a modified objective function $J(w;\alpha)$ that is differentiable. In order to be applied to DDNCID [6], it is crucial that we have a good evaluation of the gradient vector $\nabla J(w;\alpha)$ and Hessian matrix $H(w;\alpha)$.

In DDNCID [6], the algorithm uses a finite difference approach to approximate the Hessian of a general function as follows:

```
function [H] = chooseHess(X,fun)
h=sqrt(eps);
n=length(X); %find the dimension of X
Id=eye(n);
[grad]=chooseGrad(X,fun);
for s=1:n
    [grad1]=chooseGrad(X+h*Id(:,s),fun);
    H(:,s)= (grad1-grad)/h;
end
H=0.5*(H+H');%average to ensure that H is symmetric
```

Though the finite difference method can provide a good approximation if the function does not change dramatically, it might not be able to provide a good approximation of Gradient and Hessian for our logarithmic barrier function.

Thus, we calculate the gradient ∇P and the Hessian $d_2 P$ exactly as follows: let $Q = \{i \in Z \,|\, (w_{low})_i \neq -\infty \text{ or } (w_{up})_i \neq \infty\}$. Set

$$\nabla P_i = \begin{cases} \alpha\left(-\frac{1}{w_i-(w_{low})_i} + \frac{1}{-w_i+(w_{up})_i}\right), & \text{if } i \in Q \\ 0, & \text{if } i \notin Q \end{cases}$$

$$\text{and } d_2P(w;\alpha) = \begin{cases} \alpha\left(\frac{1}{(w_i-(w_{low})_i)^2} + \frac{1}{((w_{up})_i-w_i)^2}\right), & \text{if } i \in Q \\ 0, & \text{if } j \neq i \text{ or } i \notin Q. \end{cases}$$

3.2 Improvements of DDNCID in Image Registration

Despite the introduction of the logarithmic barrier function [7], we also modified the algorithms in FAIR [1] and DDNCID [6] by replacing double descent with the Gauss Newton method existing in FAIR.

In the local search process of DDNCID [6], the search for a descent direction is expensive, regardless whether the direction is chosen by double descent or gradient descent. Therefore, we prefer to use the Gauss-Newton scheme provided by FAIR [1] since it is more efficient. We calculate the Hessian of the objective function $J(w;\alpha)$ by combining finite differences and direct calculation.

Regarding the term $D(w)$, the gradient $\nabla D(w)$ can be evaluated during the call of the objective function. But FAIR [1] only provides a Gauss-Newton approximation of the Hessian in the term $D(w)$ in our objective function. It rewrites the objective function as $J(w) = \psi(r(w))$, where $r = T(\boldsymbol{y}(w, \boldsymbol{x}) - R(\boldsymbol{x})$ and $\psi = \frac{1}{2} r^T r$, and optimizes a quadratic function $\hat{J}(w + dw) = J + dJ\,dw + \frac{1}{2}\,dw^T\,H\,dw \approx J(w + dw)$. The approximation error $\hat{J} - J$ is small when dw is small. Also, the approximate Hessian $H = dr^T d^2\psi\, dr$ is obviously always symmetric positive semi-definite.

However, DDNCID [6] needs a negative curvature to perturb the points around saddle points and maxima and, thus, to escape to another critical point. We also seek a reliable method of calculating the Hessian of the term for distance measure.

Furthermore, if a point w is close to the boundary, $S(w + \Delta w; \alpha)$ could be infinite, causing the evaluation of $\nabla J(w;\alpha)$ and $H(w;\alpha)$ to be inaccurate.

Thus, we modify the evaluation of Gradient and Hessian of the objective function $J(w)$ as follows.

1. Use the finite difference to evaluate the Hessian of $D(w)$, which we denote as $d_2D(w)$.

Note: In the case that $w + \Delta w$ is out of boundary, we instead evaluate

$$d_2 D(w) = \frac{\nabla D(w + \Delta w) - \nabla D(w)}{h}.$$

2. Calculate the exact gradient $\nabla P(w; \alpha)$ and the Hessian $\nabla P_2(w; \alpha)$ of the barrier function $P(w; \alpha)$.

3. Combine the two terms to get the gradient $\nabla J(w; \alpha)$ and the Hessian $J_2(w; \alpha)$:

$$\nabla J(w; \alpha) = \nabla D(w) + \nabla P(w; \alpha), \text{ and} \tag{3.4}$$

$$J_2(w; \alpha) = d_2 D(w) + d_2 P(w; \alpha). \tag{3.5}$$

In some test cases, we discover that the first implementation of Gauss-Newton finds the first critical point next to the boundary. This is undesirable. Since the perturbation is based on the largest eigenvalues of the Hessian of that critical point, the escape process may add a color noise that is not feasible. Under such circumstances, we prefer Gradient Descent to find our first minimum.

We set up log barrier constraints for rigid and affine transformation models to control the extreme shift when solving the image registration problem. In the future, we will consider more image models to test the reliability of our new objective function with the logarithmic function. Also, KKT [8] should be a direction for future work as it is able to find minimizers around the boundary.

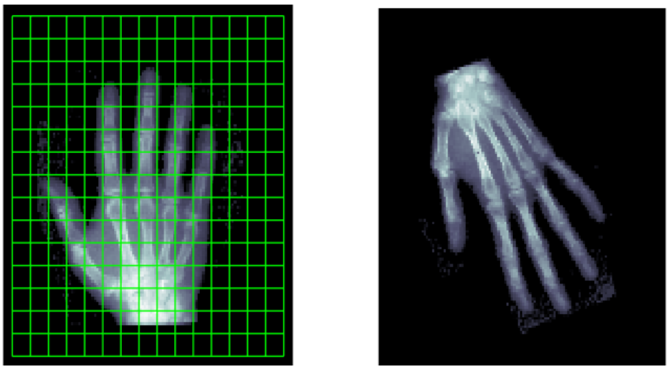
Chapter 4

Numerical Experiments

In this Chapter, some numerical results are presented.

4.1 2D Affine Registration without Boundary Constraints

Let us consider the reference image and the template image as shown in Figure 4.1. In this experiment, the domain Ω is given by $[0, 20] \times [0, 25] \in \mathbb{R}^2$



(a) Reference Image and regular grid xc (green)

(b) Template Image

Figure 4.1: Affine Image Registration Problem

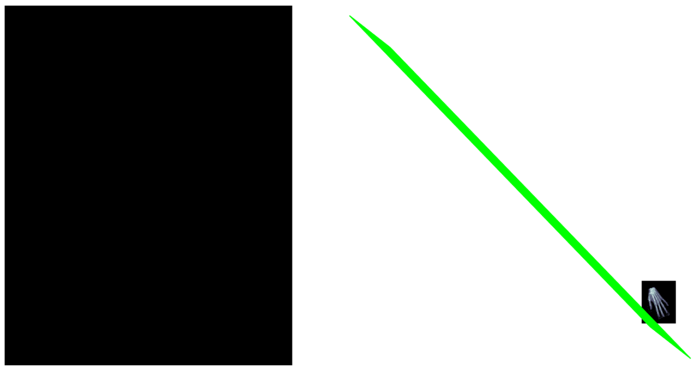
In Table 4.1 , we provide an array of W_{min}. Each column represents a specific minimum. Some local minima contain extremely large entries of w_3 and w_6 when compared to our domain. These two entries w_3 and w_6 correspond to the shifting of the template image in affine transformation.

To visualize an undesired transformation, we pick the sixth minimum in the array. In Figure 4.2, we see a possible undesired result obtained by applying the unconstrained version

w_1	0.62	0.03	-1.08	0.61	-0.00	1.86	1.46	-3.13	-1.27	2.57
w_2	-0.35	-0.13	-6.92	2.92	-0.79	1.03	-0.35	-7.82	-7.52	1.38
w_3	4.49	-0.87	31.89	-1.87	-1.26	-18.96	28.05	10.32	35.41	-28.26
w_4	0.67	0.02	1.33	-4.11	-0.00	8.81	2.41	0.79	1.00	11.38
w_5	0.62	0.04	10.31	-6.58	0.58	1.59	17.17	-3.56	7.79	2.90
w_6	3.93	-3.01	-36.72	-0.09	0.93	-84.44	-61.64	0.59	-27.32	-117.55

Table 4.1: An array of minimizers W_{min} solved by DDNCID without boundary constraints

of the method. Although we were able to find some minima, they could be far away from our domain of interest. Figure 4.2 shows our need to introduce boundary constraints on our domain.

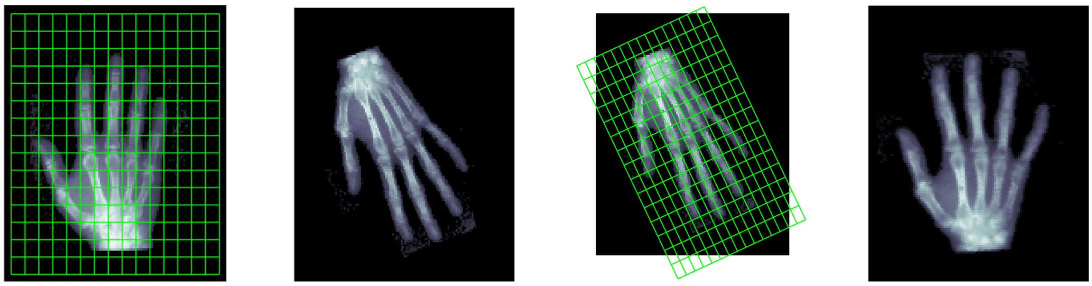


(a) Transformed Template

(b) Template Image and Transformed Cell Grids yc

Figure 4.2: Affine Image Registration Problem

4.2 2D Affine Registration with Boundary Constraints

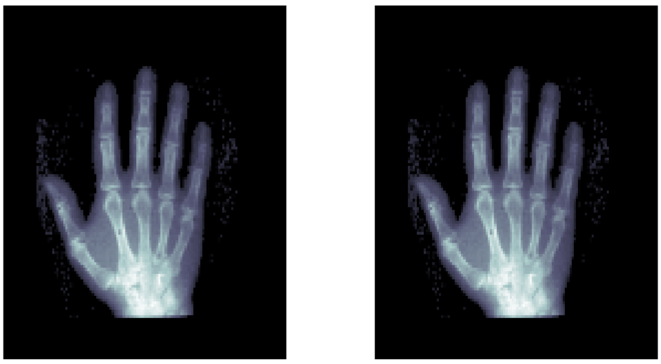


(a) Reference Image and regular grid (green) (b) Template Image (c) Template Image and Transformed Grid (green) (d) Transformed Template

Figure 4.3: Affine Registration Problem solved by Global Optimization

To test the applicability of affine registration, as we mentioned in Chapter 1, we set up a problem that is not solved by the Gauss-Newton scheme. Using the new algorithm to solve this registration problem, we have found 76 critical points and 39 among them are local minimizers.

4.3 2D Rigid Registration



(a) Reference Image (b) Template Image

Figure 4.4: Rigid Registration Problem

In 2D rigid registration, we set up the problem by using the same image for both reference image and template image. Thus, we know the global solution to be $w_c = [0; 0; 0]$, as no shifting and no rotation are needed to transform an image to itself. We test our DDNCID with 12 starting guesses. For each initial guess, we start with the Gauss-Newton scheme as the local optimization method. If the first step minimization results in a minimizer close to our constraints, we use Gradient Descent instead of Gauss-Newton in our first step to minimize the initial guess.

Here, we provide the registration result of DDNCID with the initial starting guess $w_0 = [-1.5708; -5.3333; 4.0000]$.

(a) Initial Guess $w_0 = [-1.5708; -5.3333; 4.0000]$

(b) Transformed Template

Figure 4.5: Rigid Registration Problem with initial guess $w_0 = [-1.5708; -5.3333; 4.0000]$

In this experiment, we set the maximum number of iteration to be 120. With an additional initial local minimization process, DDNCID finds 121 critical points, of which 66 points are local minima. As shown in Figure 4.5, our method is able to find the global minimum.

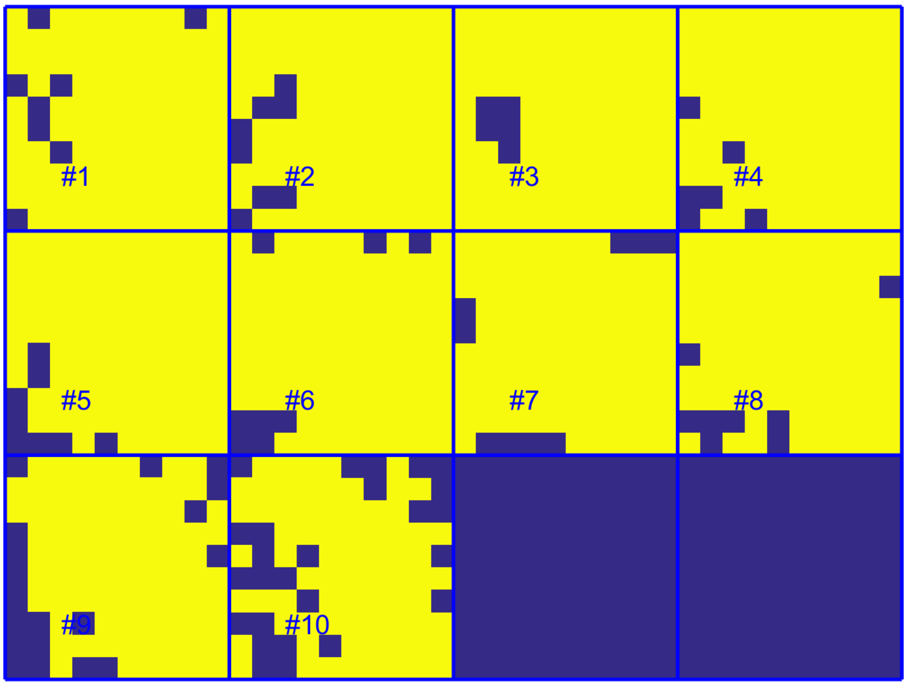


Figure 4.6: Results of Large Experiment in Rigid Transformations

To enlarge our test cases, we increase the number of starting guesses to 1000. These starting guesses are equally spaced points in $[-\pi, -8, -8] \times [\pi, 8, 8]$. In Figure 4.6, each numbered square (from 1 to 10) displays the results of 100 cases. A success of the global optimization method is marked as a small yellow square in each numbered square, while a failure is represented by blue. As we can tell from the results, our log barrier constraint provides accurate solutions for a majority of our initial guesses. Failures of some initial guesses often occur when we start with a local minimum near the boundary or we need a larger number of iterations.

Chapter 5

Conclusion

In this chapter, we summarize our current work and discuss potential future directions.

5.1 Current Work

In this thesis, we test the applicability of a global optimization method in image registration problems. To enable the implementation, we extend the current version of DDNCID by adding barrier constraint into the objective function to avoid undesirable transformations. To avoid misclassification of critical points, we also modify the evaluation of Hessian, using a combination of finite difference of the distance measure and direct calculation of the log barrier constraint. Also, for efficiency purposes, we replace Double Descent with the Gauss-Newton scheme or the Gradient Descent method.

Our updated DDNCID successfully solves the image registration problem that could not be achieved by the Gauss-Newton method. Numerical experiments have been tested on both affine and rigid transformations. With different starting guesses, global optimization can find several critical points and local minima, which includes the global minimum in most cases.

5.2 Future Directions

In terms of future work, we consider adding the entirely direct calculation of the Hessian of the objective function. We also look forward to applying the updated DDNCID to higher dimensional data as well as to transformations with a larger number of unknowns. Additionally, we expect to optimize the algorithms to foster the efficiency of our method.

Appendix A

MATLAB Code

A.1 Log Barrier Constraint

```
function [Sc,logdS,logd2S]=getLogBarrier(uc,omega,m,varargin)
%set up parameter
alpha        = 1;
low          = -Inf;
high         =Inf;

for k=1:2:length(varargin),      % overwrites default parameter
  eval([varargin{k},'=varargin{',int2str(k+1),'};']);
end;

if all(low==-Inf)&& all(high==Inf)
      Sc=0;
    logdS=0;
    logd2S=0;
    return;
end;
if alpha==0
    Sc=0;
    logdS=0;
    logd2S=0;
    return;
end
    n =numel(uc);
```

```
    %add penalty to boundary condition
if any(uc<=low) || any(uc>=high)
    Sc     = Inf;
    logdS  = Inf;
    logd2S = [];
%     warning('out of bound\n');
    return
end

% allocate output
Sc     = 0.0;
logdS  = zeros(n,1);
logd2S = zeros(n,1);

% find indices of components bounded above and below
id = (low > -Inf) & (high < Inf);
B   = -log(uc( id)-low(id))-log(high(id)-uc(id));
Sc  = Sc + sum(B(:));
logdS(id) = -1./(uc(id)-low(id))+1./(high(id)-uc(id));
d2YLow       = (uc(id)-low(id)).^2;
d2YHigh      = (high(id)-uc(id)).^2;
logd2S(id) = (1./d2YLow)+(1./d2YHigh);

% find indices of components only bounded above
id = (low == -Inf) & (high < Inf);
B   = -log(high(id)-uc(id));
Sc  = Sc+  sum(B(:));
logdS(id) = 1./(high(id)-uc(id));
d2YHigh      = (high(id)-uc(id)).^2;
logd2S(id) = (1./d2YHigh);

% find indices of components only bounded below
id = (low > -Inf) & (high == Inf);
B   = -log(uc(id)-low(id));
Sc  = Sc + sum(B(:));
logdS(id) = -1./(uc(id)-low(id));
d2YLow       = (uc(id)-low(id)).^2;
logd2S(id) = (1./d2YLow);
```

```
Sc = alpha*Sc;
logdS = alpha* logdS';
logd2S=alpha * diag(logd2S);
return;
```

A.2 Hessian Approximation

```
function[id,sigma,H,L]=findsigmafull(x,fun)
regularizer('set','c',0);% H1=d2P from log barrier
[~,~,~,H1]=fun(x);
regularizer('set','c',1);
a=regularizer('get','alpha');
regularizer('set','alpha',0);
H2=chooseHess(x,fun);
regularizer('set','alpha',a);% H2=d2J from distance measure
H=H1+H2;
[V,D]=eigs(H,2,'be');
lambdabig=D(2,2); lambdasmall=D(1,1);
vbig=V(:,2); vsmall=V(:,1);
% check id by comparing the eigenvalues
if (sign(lambdasmall)==sign(lambdabig))
    if lambdasmall>0
        id=1;
    else
        id=-1;
    end
else
    id=0;
end
if id==1
    sigma=vbig*vbig';
    L=lambdabig;
else
    sigma=vsmall*vsmall';
    L=lambdasmall;
end
```

```
function [H] = chooseHess(X,fun)
    h=sqrt(eps);
    % %      h =10;
    n=length(X);
%      h = 1e-5;
    Id=eye(n);
    [grad]=chooseGrad(X,fun);
    for s=1:n
        if size(X,1)==3 % && s==1
            [grad0]=chooseGrad(X+h*Id(:,s),fun);
            H(:,s)= (grad0-grad)/h;
        else
            [grad1]=chooseGrad(X+h*Id(:,s),fun);
            if grad1~=Inf
                H(:,s)= (grad1-grad)/h;
            else
                [grad2]=chooseGrad(X-h*Id(:,s),fun);
                H(:,s)= (grad2-grad)/h;
            end
            %point infeasible x-h
        end

    end
    H=0.5*(H+H');
    end
```

Bibliography

[1] J. Modersitzki. *FAIR: Flexible Algorithms for Image Registration*. Society for Industrial and Applied Mathematics, Philadelphia, print edition, 2009.

[2] B. Fischer and J. Modersitzki. Ill-posed medicine—an introduction to image registration. *Inverse Problems*, 24(3):1–16, 2008. URL http://stacks.iop.org/0266-5611/24/i=3/a=034008.

[3] A. Goshtasby. *2-D and 3-D image registration for medical, remote sensing, and industrial applications*. J. Wiley Sons, 2005.

[4] D. L. G. Hill, P. G. Batchelor, M. H. Holden, and Hawkes D. J. Medical image registration. *Physics in Medicine Biology*, 46(3):R1, 2001. URL http://stacks.iop.org/0031-9155/46/i=3/a=201.

[5] J. Hadamard. *Sur les problèmes aux dérivées partielles et leur signification physique*. Princeton University Bulletin, 1902.

[6] L. Dieci, M. Manetta, and H. Zhou. Double Descent and Intermittent Color Diffusion for Global Optimization and landscape exploration. *ArXiv e-prints*, April 2017. URL https://arxiv.org/abs/1706.05285.

[7] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer series in operations research. Springer, New York, 1999. ISBN 0387987932.

[8] A. Beck. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2014. ISBN 1611973643, 9781611973648.