Direct Optimization of the Deformable Registration Problem

Patrick Wucherer
July 2007

Abstract. In this report we discuss two different direct approaches for solving the optimization problem of the deformable registration in 2D. The aim is, to minimize the nonlinear least squares cost function [1]:

$$\mathfrak{F}(u) = \frac{1}{2}\sum_i (T(x_i + u_i) - R(x_i))^2 + \alpha\frac{1}{2}\sum_i (||\nabla u_i^{(x)}||^2 + ||\nabla u_i^{(y)}||^2)$$

The first part of the function is the sum of squared differences and the second part is a component for penalizing the function, if the displacement field is not smooth.
We tried to solve this problem with direct techniques, so we first discretize the cost function and then solve it with the Gradient Descent and the Gauss-Newton minimization method [3, 9]. We compare the two methods by applying them to synthetic images, to show their different convergence. We conclude, that the Gauss-Newton method should be preferred, because of the faster convergence, not only in respect to the number of iterations, but also in respect to time in our implementation.

1 Introduction

The Deformable Registration is one of the major problems of image processing. Especially in medicine there are several application fields such as, making preoperative data similar to intraoperative data for adapting the planning, which is previously done by the surgeon on preoperative data. In this report the results are based on mono-modal images, that means that the images are taken by the same imaging machinery. The aim is to deform one image called template T in such a way, that it becomes similar to another image called reference R. The underlying idea is to have a function $\mathfrak{F}$ with two main components: the first one is a distance measure D between two images and the second one is a regularisation term S.; see e.g.[2]. The variational approach is to find a displacement field u that minimizes the function

$$\mathfrak{F}[u] := D(R, T, u) + \alpha S(u) \tag{1}$$

α is used as a regularization parameter greater 0 to balance the similarity measure against the regularization term [1, 2]. It is chosen by testing. For the sake of simplicity, we focus on

the differentiable Sum of Squared Differences (SSD) measure. We assume that two images are approximatly the same with respect to the intensity, so that the intensity residual is appropriate to describe their misalignment. The used regularization term consists of the gradient of the displacement field u. The regularization term incorporates smoothness constraints in the displacement field. There are other possibilities of regularization term for image registration, which are based on physical approaches to describe elasticity, fluid or diffusion, see [8].

2 Discretization

2.1 Template and Reference

The reference R and the template T can be seen as vectors $R = (r_1, ..., r_{nm})$ and $T = (t_1, ..., t_{nm})$, which have the size nm, where n is the imagesize in the x-dimension and m in y-dimension.

2.2 Displacement Field

Then we define a displacement field u, containing the values of the x-dimension in $u^{(x)} \in \mathbb{R}^{nm}$ and of the y-dimension in $u^{(y)} \in \mathbb{R}^{nm}$.

$$u = \begin{bmatrix} u^{(x)} & u^{(y)} \end{bmatrix}; u \in \mathbb{R}^{nm \times 2} \tag{2}$$

where $u_i^{(x)} \in \mathbb{R}$ represents the value for the displacement in x-dimension at point i, and $u_i^{(y)} \in \mathbb{R}$ for the displacement in y-dimension. $i \in \{1, ..., nm\}$.

2.3 Sum of Squared Differences

The similarity SSD $D : \mathbb{R}^{nm} \times \mathbb{R}^{nm} \times \mathbb{R}^{nm} \to \mathbb{R}$ is

$$D(R, T, u) = \frac{1}{2} \sum_i (T(w(x_i, u_i)) - R(x_i))^2 \tag{3}$$

where x_i is a particular position and w is a function, which warps x_i with the displacement u_i.

2.4 Regularization Term

We describe the discretization of the regularization term S, which consists of the gradient of the displacement field as followed. Assume for the regularization term $S : \mathbb{R}^{nm \times 2} \to \mathbb{R}$

$$S(u) = \frac{1}{2} \sum_i (||\nabla u_i^{(x)}||^2 + ||\nabla u_i^{(y)}||^2) \tag{4}$$

3 Gradient Descent

The Gradient Descent method is a standard method for minimizing non-linear systems. The update value is the negative gradient of the function $\mathfrak{F}(\mathfrak{u})$. At every step it is going in a descent direction until it reaches the minimum of the function $\mathfrak{F}$. Therefore we need the first

derivative of the function $\mathfrak{F}(\mathfrak{u})$. We start with finding the function $f(u)$ as described in [3]. First of all, we can write the similarity measure as:

$$D(R,T,u) = \frac{1}{2}(T(w(x,u)) - R(x))^\top (T(w(x,u) - R(x)))$$

and the regularization term as:

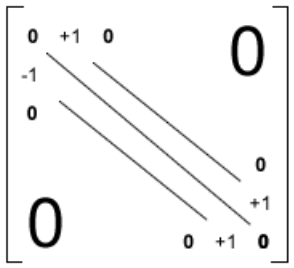
$$S(u) = \frac{1}{2}\begin{bmatrix} G_x u^{(x)} \\ G_y u^{(x)} \end{bmatrix}^\top \begin{bmatrix} G_x u^{(x)} \\ G_y u^{(x)} \end{bmatrix} + \frac{1}{2}\begin{bmatrix} G_x u^{(y)} \\ G_y u^{(y)} \end{bmatrix}^\top \begin{bmatrix} G_x u^{(y)} \\ G_y u^{(y)} \end{bmatrix}$$

The least squares problem is stated as $F(u) = f(u)^\top f(u)$ [3, 9]. We use the equations above to get the function $f(u)$

$$f(u) = \begin{pmatrix} T(w(x,u)) - R(x) \\ \sqrt{\alpha} G_x u^{(x)} \\ \sqrt{\alpha} G_y u^{(x)} \\ \sqrt{\alpha} G_x u^{(y)} \\ \sqrt{\alpha} G_y u^{(y)} \end{pmatrix} \tag{5}$$

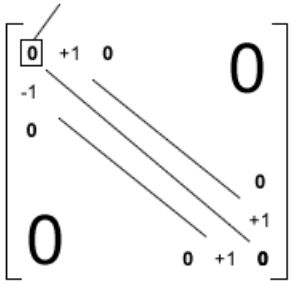
The update is a vector $u \in \mathbb{R}^{nm\times 2}$ and G_x and G_y are elements of $\mathbb{R}^{nm\times nm}$ denote the derivation operators.

$G_x =$



Dimension: half the dimension of vector v

$G_y =$



The vector is the linearized $n \times m$ matrix, of which the gradient should be calculated. Actually the boundary conditions are not included in these operators.

When we put it altogether we get:

$$\begin{aligned}
\mathfrak{F}(u) &= \frac{1}{2} f(u)^\top f(u) && (6)\\
&= \begin{pmatrix} T(w(x,u)) - R(x) \\ \sqrt{\alpha} G_x u^{(x)} \\ \sqrt{\alpha} G_y u^{(x)} \\ \sqrt{\alpha} G_x u^{(y)} \\ \sqrt{\alpha} G_y u^{(y)} \end{pmatrix}^\top \begin{pmatrix} T(w(x,u)) - R(x) \\ \sqrt{\alpha} G_x u^{(x)} \\ \sqrt{\alpha} G_y u^{(x)} \\ \sqrt{\alpha} G_x u^{(y)} \\ \sqrt{\alpha} G_y u^{(y)} \end{pmatrix} && (7)\\
&= \frac{1}{2} \sum_i (T(w(x_i, u_i)) - R(x_i))^2 + \alpha \frac{1}{2} \sum_i (||\nabla u_i^{(x)}||^2 + ||\nabla u_i^{(y)}||^2) && (8)
\end{aligned}$$

The calculation of the gradient of the function $f(u) \in \mathbb{R}^{nm5}$ is stated below. As we know that, the update vector is $u \in \mathbb{R}^{nm \times 2}$, the gradient must have the dimension $\mathbb{R}^{2nm \times nm5}$.

$$\frac{\partial}{\partial u} f(u)^\top = \begin{bmatrix} \frac{\partial}{\partial u^{(x)}} f(u) \\ \frac{\partial}{\partial u^{(y)}} f(u) \end{bmatrix} = J(u)^\top = \begin{pmatrix} \frac{\partial}{\partial x} T(w(x,u))^\top & -\sqrt{\alpha} G_x & -\sqrt{\alpha} G_y & \mathbf{0} & \mathbf{0} \\ \frac{\partial}{\partial y} T(w(x,u))^\top & \mathbf{0} & \mathbf{0} & -\sqrt{\alpha} G_x & -\sqrt{\alpha} G_y \end{pmatrix} \quad (9)$$

The operator matrix is $-G$, because of $G^\top = -G$. The derivation operator is antisymmetric and has the value ± 1 on the respective diagonals, see above. The matrix $\frac{\partial}{\partial x} T(w(x,u)) \in \mathbb{R}^{n \times m}$ contains on the diagonal the values of $\frac{\partial}{\partial x} T(w(x,u))$.

$$\begin{aligned}
\mathfrak{F}'(u) &= J(u)^\top f(u) && (10)\\
&= \begin{pmatrix} \frac{\partial}{\partial x} T(w(x,u))^\top & -\sqrt{\alpha} G_x & -\sqrt{\alpha} G_y & \mathbf{0} & \mathbf{0} \\ \frac{\partial}{\partial y} T(w(x,u))^\top & \mathbf{0} & \mathbf{0} & -\sqrt{\alpha} G_x & -\sqrt{\alpha} G_y \end{pmatrix} \begin{pmatrix} T(w(x,u)) - R(x) \\ \sqrt{\alpha} G_x u^{(x)} \\ \sqrt{\alpha} G_y u^{(x)} \\ \sqrt{\alpha} G_x u^{(y)} \\ \sqrt{\alpha} G_y u^{(y)} \end{pmatrix} && (11)\\
&= \begin{pmatrix} \frac{\partial}{\partial x} T(w(x,u))(T(w(x,u)) - R(x)) - \alpha G_x G_x u^{(x)} - \alpha G_y G_y u^{(x)} \\ \frac{\partial}{\partial y} T(w(x,u))(T(w(x,u)) - R(x)) - \alpha G_x G_x u^{(y)} - \alpha G_y G_y u^{(y)} \end{pmatrix} && (12)\\
&= \begin{pmatrix} \frac{\partial}{\partial x} T(w(x,u))(T(w(x,u)) - R(x)) - \alpha (G_x G_x u^{(x)} + G_y G_y u^{(x)}) \\ \frac{\partial}{\partial y} T(w(x,u))(T(w(x,u)) - R(x)) - \alpha (G_x G_x u^{(y)} + G_y G_y u^{(y)}) \end{pmatrix} && (13)\\
&= \begin{pmatrix} \frac{\partial}{\partial x} T(w(x,u))(T(w(x,u)) - R(x)) - \alpha \Delta u^{(x)} \\ \frac{\partial}{\partial y} T(w(x,u))(T(w(x,u)) - R(x)) - \alpha \Delta u^{(y)} \end{pmatrix} && (14)
\end{aligned}$$

Where Δ is the laplacian operator, which is the sum of the second derivatives in x and y direction.

To ensure, that we are calculating the correct gradient of $F(u)$, we have a closer look at the first and the second part of the cost function. First we derive the derivative of the SSD:

$$\frac{\partial}{\partial u} D(R,T,u) = \nabla T(w(x,u))^\top (T(w(x,u)) - R(x))$$

Secondly the regularization term:

$$S(u) = \frac{1}{2} \sum_i ||\nabla u_i^{(x)}||^2 + ||\nabla u_i^{(y)}||^2$$

The regularization term is treated separately in the x and y direction.

$$S(u) = \frac{1}{2} S^{(x)}(u^{(x)}) + \frac{1}{2} S^{(y)}(u^{(y)})$$

We show the derivative in one direction. It is the same in the other direction.

$$\begin{aligned}
S^{(x)}(u^{(x)}) &= \frac{1}{2} \sum_i ||\nabla u_i^{(x)}||^2 && (15)\\
&= \frac{1}{2} \sum_i (\partial_x u^{(x)})^2 + (\partial_y u^{(x)})^2 && (16)\\
&= \frac{1}{2} (G_x u^{(x)})^\top (G_x u^{(x)}) + \frac{1}{2} (G_y u^{(x)})^\top (G_y u^{(x)}) && (17)\\
&= \frac{1}{2} (u^{(x)})^\top G_x^\top G_x u^{(x)} + \frac{1}{2} (u^{(x)})^\top G_y^\top G_y u^{(x)} && (18)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial S^{(x)}}{\partial u^{(x)}}(u^{(x)}) &= \frac{1}{2} (2 G_x^\top G_x u^{(x)} + 2 G_y^\top G_y u^{(x)}) && (19)\\
&= (G_x^\top G_x + G_y^\top G_y) u^{(x)} && (20)\\
&= -(G_x G_x + G_y G_y) u^{(x)} && (21)\\
&= -\Delta u^{(x)} && (22)
\end{aligned}$$

Finally we are sure, that we have adapted our cost function correctly to the least squares problem and can minimize it with conventional methods.

With the Gradient Descent method the update minimizes the function locally. The convergence of this method is at the final stage linear and often very slow [3,9]. The method is based on the Taylor Series [3,9]:

$$\mathfrak{F}(u + \gamma h) = \mathfrak{F}(u) + \gamma h^\top \mathfrak{F}'(u) + O(\gamma^2) \simeq \mathfrak{F}(u) + \gamma h^\top \mathfrak{F}'(u) \text{ for } \gamma \text{ sufficiently small} \quad (23)$$

h is a descent direction for $\mathfrak{F}$ at x if $h^\top \mathfrak{F}'(u) < 0$. The update vector is just $h = -\mathfrak{F}'(u)$. The timestep γ is chosen by testing.

Algorithm 1 Pseudocode Gradient Descent Method

1: Compute $h = -J(u)^\top f(u)$
2: $u := u + \gamma h$

4 Gauss-Newton

The second method discussed in this report is the Gauss-Newton method. It is possible to achieve quadratic convergence even though the second derivative is not calculated, but only in special cases [3,9]. The method is based on the first derivatives of the vector function. Based on the linear approximation of the function f we get the Taylorseries:

$$f(u+h) = f(u) + J(u)h \tag{24}$$

From equation 7 we get

$$\begin{aligned} \mathfrak{F}(u+h) &\equiv \frac{1}{2}(f(u)+J(u)h)^\top(f(u)+J(u)h) && (25)\\ &= \frac{1}{2}f(u)^\top f(u) + h^\top J(u)^\top f(u) + \frac{1}{2}h^\top J(u)^\top J(u)h && (26)\\ &= \mathfrak{F}(\mathfrak{u}) + h^\top J(u)^\top f(u) + \frac{1}{2}h^\top J(u)^\top J(u)h && (27) \end{aligned}$$

where, h is the Gauss-Newton step, which minimizes the function $\mathfrak{F}(u+h)$

$$h = \text{argmin}_h\{\mathfrak{F}(u+h)\} \tag{28}$$

The derivative of the linearisation is:

$$\mathfrak{F}'(u+h) = J^\top(u)f(u) + J(u)^\top J(u)h \tag{29}$$

$J(u)^\top J(u)$ as an approximation of the Hessian matrix and is positive definite, hence we know that $\mathfrak{F}(u+h)$ has a unique minimizer, see [3,7] .

$$\begin{aligned} J(u)^\top J(u)h &= -J^\top(u)f(u) && (30)\\ h &= -(J(u)^\top J(u))^{-1}J^\top(u)f(u) && (31) \end{aligned}$$

Since $h^\top\mathfrak{F}'(u) = -J^\top(u)f(u) = -h^\top(J(u)^\top J(u))h < 0$ this is a decent direction for $\mathfrak{F}(u)$. In summary the algorithm we use to get the update h is:

Algorithm 2 Pseudocode Gauss-Newton

1: Compute $h = -J(u)^+ f(u)$,where $J(u)^+ = (J(u)^\top J(u))^{-1}J(u)^\top$
2: $u := u + \gamma h$

γ is chosen by testing.

5 Comparison GD vs. GN

The image set 1 consists of synthetical images. The parameters for the Gradient Descent method were $\alpha = 0.1$ and $\gamma = 1$. We ran this method 1500 iterations long and compared it to the Gauss-Newton method, which ran for 350 iterations and had $\alpha = 0.1$ and $\gamma = 0.8$. The methods were tested on a 2.13 Core2 with 2 GB Ram.

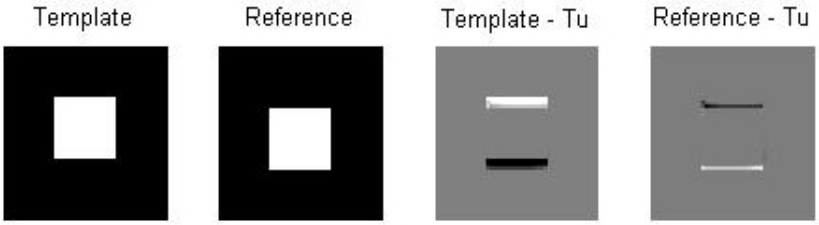


Figure 1: The difference between the template and the reference was minimized, which you can clearly see on behalf of the differences between template - deformed template and reference - deformed template

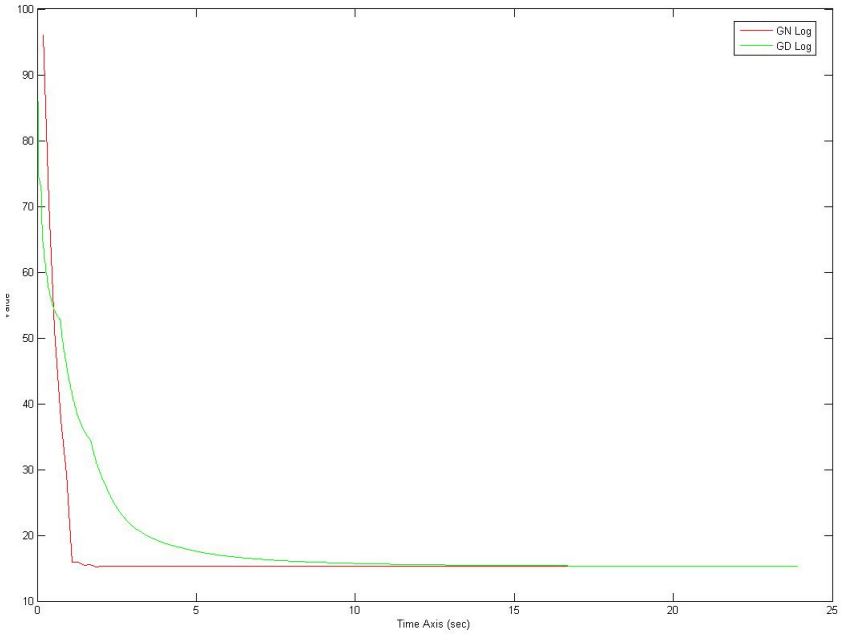


Figure 2: Gradient Descent vs. Gauss-Newton

6 Appendix

6.1 Update

We have a closer look at our update. Based on [12], we combine the two subsequent displacement fields in the Eulerian Reference Frame as followed. Naming the displacement field at timepoint t as u^t and at timepoint t+1 as u^{t+1}, then the simple addition of these two displacement fields only presents an approximation, because the underlying image for the calculation of the displacement has changed and therefor the values of the displacement at the position x differs, where x is a particular position of the values of the displacement field u^t and u^{t+1} . When we have calculated the displacement for the current image and want to add it to the previous displacement the following formula does the right thing:

$$u = warp(u^t, u^{t+1}) + u^{t+1}$$

Just a short description how to take this into account in an algorithm.

Algorithm 3 Pseudocode Correct Displacement

1: First calculate the values for minimizing the function.
2: Secondly, warp the displacement field with these values.
3: At last, add the displacement field values to the calculated values.

The advantage of this is, that it is better for larger timesteps, but it is quite time consuming.

6.2 Gaussian Pyramid

Motivation of the gaussian pyramid is to find an adequate solution for the displacement field on a coarse level and then prolongate it to a finer level. By doing this, the optimal displacement field is prolongated through all levels of the gaussian pyramid and in the end we have a good result in less time. First a low pass filter is applied and the amount of pixels is halved. Doing that iteratively we get the gaussian pyramid. For the prolongation of the displacement field from a coarser level to a finer level the bilinear interpolation is used.

Algorithm 4 Gaussian Pyramid

1: Reduce: Image is filtered by a Gauss filter and the pixel amount is halved.
2: Expand: Increasing the size of the displacement field is done by using the bilinear interpolation method and the values are multiplied by 2.

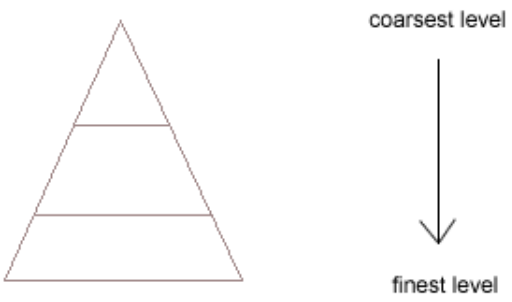


Figure 3: Gaussian Pyramid

References

[1] Eldad Haber, J. Modersitzki; A Multilevel Method for Image Registration

[2] J. Modersitzki, Numerical Methods for Image Registration, Oxford University Press, New York 2004

[3] K. Madsen, H.B. Nielsen, O.Tingleff; Methods for non-linear least squares problems, 2nd Edition, April 2004,

[4] Fetzer, Fränkel; Mathematik 1 Auflage 6; Springer Verlag 2000

[5] Günter Gramlich, Wilhelm Werner; Numerische Mathematik mit Matlab; dpunkt. verlag 2000

[6] Jan Modersitzki; Numerical Methods for Image Registration; Oxford Science Publications 2004

[7] Huckle, Schneider; Numerik für Informatiker; Springer Verlag 2002

[8] M. Knorrenschild; Numerische Mathematik; Fachbuchverlag Leipzig 2003

[9] Poul Erik Frandsen, Kristian Jonasson, Hans Bruun Nielsen, Ole Tingleff, Unconstrainted Optimization, 3. Edition March 2004