SARAJEVO JOURNAL OF MATHEMATICS
Vol.7 (19) (2011), 135–144

APPLICATION OF AN IMAGE REGISTRATION METHOD TO NOISY IMAGES

M. A. AKINLAR AND R. N. IBRAGIMOV

Abstract. The purpose of this article is twofold: First to overview the recently proposed 3D image registration algorithm presented in the Ph.D. dissertation of the first author and secondly to apply the results to the registration of images which have a certain level of noise. Our method is developed by adjusting the divergence and curl of the image displacement field by means of which we can control translation, rotation, and deformations of image pixels. Our method incorporates sum of squared differences as the similarity metric and uses the Lagrange multipliers method to solve the existing optimization problem from which we obtain an optimality system that consists of four Poisson equations. In the 2D case a finite-difference multigrid strategy is used to solve these Poisson equations. Multi-level coarse-to-fine iterations allow us efficient, accurate and robust registration even if one or both of the images to be registered have a significant level of noise.

1. Introduction

In basic terms, image registration can be defined as a process of determining the optimal transform that maps points from one image to the corresponding points in another image. In general the image registration problem can be formulated as an optimization problem:

$$\mathcal{J}[\mathbf{R}, \mathbf{T}; u] := \min \Big\{ C_{sim} + \beta C_{reg} \Big\}, \tag{1.1}$$

where C_{sim} is the similarity metric between template $\mathbf{T}$ and reference images $\mathbf{R}$, and C_{reg} is the regularization term due to cracks, folding or other unwanted deformations and β is the regularization constant. We can separate the image registration problem to three main components as **transformation models** (Rigid, non-rigid, hybrid), **Similarity metrics** (Intensity-based, geometry-based) and **Optimization procedure** (Lagrange multipliers method, gradient descent method, etc.) Detailed treatment for different

2000 *Mathematics Subject Classification.* 65D18, 65J05, 97N40.
Key words and phrases. Image registration, image noise, Lagrange multipliers.

image registration techniques and some applications of it can be seen in ([1], [3]) and in the references therein.

In his Ph.D. dissertation [2] the first author introduced a 3D image registration method and applied this method to the registration of noise-free images. In this paper we illustrate the results obtained upon applying this image registration method to the registration of images which contain a certain level of noise.

This paper is organized as follows: In the first section we briefly mention the methodology behind image registration. The second section introduces our image registration method. In the remaining three sections we introduce the noise concept, present computational examples and finish the paper with a summary and conclusion.

2. Our method for image registration

Given a *reference image* $\mathbf{R}(\mathbf{x})$ and a *template image* $\mathbf{T}(\mathbf{x})$, the image registration problem can be cast as an optimization problem which is defined as follows: find a mapping $\boldsymbol{\phi}(\mathbf{x})$ that minimizes the L^2-norm of the difference between $\mathbf{T}\big(\boldsymbol{\phi}(\mathbf{x})\big)$ and $\mathbf{R}(\mathbf{x})$ over Ω. In order to achieve this, we define the similarity functional

$$\mathcal{J}(\boldsymbol{\phi}, f, \mathbf{g}) = \frac{1}{2}\int_\Omega \left|\mathbf{T}\big(\boldsymbol{\phi}(\mathbf{x})\big) - \mathbf{R}(\mathbf{x})\right|^2 d\mathbf{x} + \frac{\omega}{2}\int_\Omega |f(\mathbf{x})|^2\, d\mathbf{x} + \sum_{i=1}^{3}\frac{w_i}{2}\int_\Omega |g_i(\mathbf{x})|^2\, d\mathbf{x}, \quad (2.1)$$

where

$$\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x}) \quad (2.2)$$

with $\mathbf{u}$ satisfying

$$\nabla\cdot\mathbf{u} = f - 1 \text{ and } \nabla\times\mathbf{u} = \mathbf{g} \text{ in } \Omega \quad \text{and} \quad \mathbf{u} = \mathbf{0} \text{ on } \partial\Omega. \quad (2.3)$$

In (2.1), ω and w_i, $i = 1, 2, 3$, are penalty weights and g_i, $i = 1, 2, 3$, denote the components of the vector $\mathbf{g}$. Let us note that the argument $\nabla\cdot\mathbf{u} = f - 1$ improves the conservation of the size. Then, we minimize $\mathcal{J}(\boldsymbol{\phi}, f, \mathbf{g})$ subject to the constraints (2.2) and (2.3).

We use the Lagrange multiplier method to transform the constrained minimization problem into an unconstrained saddle point problem. To this end, we introduce the Lagrange multipliers $q(\mathbf{x})$ and $\mathbf{v}(\mathbf{x})$ and define the Lagrangian functional

$$\mathcal{L}(\mathbf{u}, f, \mathbf{g}, q, \mathbf{v}) = \mathcal{J}\big(\boldsymbol{\phi}(\mathbf{u}), f, \mathbf{g}\big) + \int_\Omega q(\nabla\cdot\mathbf{u} - f + 1)d\mathbf{x} + \int_\Omega \mathbf{v}\cdot(\nabla\times\mathbf{u} - \mathbf{g})d\mathbf{x}.$$

The constraint (2.2) is explicitly substituted into (2.1) and the state and co-state functions are required to satisfy $\mathbf{u} = \mathbf{0}$ and $\mathbf{v} = \mathbf{0}$ on $\partial\Omega$, respectively. Saddle points of Lagrangian functional satisfy an optimality system that consists of state equations, co-state equations, and optimality conditions; the optimality system is obtained from the first-order necessary conditions for the stationarity of $\mathcal{L}$. The rigorous justification for the use of the Lagrange multiplier method and the existence of solutions of the optimality system was established in [4].

State Equations. Setting the variations of $\mathcal{L}(\mathbf{u}, f, \mathbf{g}, q, \mathbf{v})$ with respect to the Lagrange multipliers q and $\mathbf{v}$ recovers the constraint system (2.3). For example, for any variation δq, we have

$$\delta_q \mathcal{L} = \lim_{\varepsilon \to 0} \frac{d}{d\varepsilon} \mathcal{L}(\mathbf{u}, f, \mathbf{g}, q + \varepsilon\delta q, \mathbf{v}) = \lim_{\varepsilon \to 0} \frac{d}{d\varepsilon} \int_\Omega (q + \varepsilon\delta q)(\nabla \cdot \mathbf{u} - f + 1) d\mathbf{x}$$

so that invoking the first-order necessary condition $\delta_q \mathcal{L} = 0$ results in

$$\int_\Omega \delta q(\nabla \cdot \mathbf{u} - f + 1) d\mathbf{x} = 0 \quad \text{for all } \delta q$$

so that we recover the first equation in (2.3). The second equation in (2.3) is recovered in a similar manner.

Co-state equations. The co-state equations are obtained by setting the first variation of $\mathcal{L}(\mathbf{u}, f, \mathbf{g}, q, \mathbf{v})$ with respect to the state variable $\mathbf{u}$ to zero.

$$\begin{aligned}
\delta_{\mathbf{u}} \mathcal{L} &= \lim_{\varepsilon \to 0} \frac{d}{d\varepsilon} \mathcal{L}(\mathbf{u} + \varepsilon\delta\mathbf{u}, f, \mathbf{g}, q, \mathbf{v}) \\
&= \lim_{\varepsilon \to 0} \frac{d}{d\varepsilon} \Big\{ \frac{1}{2} \int_\Omega \big| \mathbf{T}\big(\mathbf{x} + \mathbf{u} + \varepsilon\delta\mathbf{u}\big) - \mathbf{R}(\mathbf{x}) \big|^2 d\mathbf{x} \\
&\qquad + \int_\Omega q\big(\nabla \cdot (\mathbf{u} + \epsilon\delta\mathbf{u}) - f + 1\big) d\mathbf{x} + \int_\Omega \mathbf{v} \cdot \big(\nabla \times (\mathbf{u} + \epsilon\delta\mathbf{u}) - \mathbf{g}\big) d\mathbf{x} \Big\} \\
&= \int_\Omega \delta\mathbf{u} \cdot \Big(\mathbf{T}_{\boldsymbol{\phi}}(\mathbf{x} + \mathbf{u}) \big(\mathbf{T}(\mathbf{x} + \mathbf{u}) - \mathbf{R}(\mathbf{x}) \big) \Big) d\mathbf{x} \\
&\qquad - \int_\Omega \delta\mathbf{u} \cdot \nabla q d\mathbf{x} + \int_\Omega \delta\mathbf{u} \cdot (\nabla \times \mathbf{v}) d\mathbf{x}
\end{aligned}$$

so that setting $\delta_{\mathbf{u}} \mathcal{L} = \mathbf{0}$, we obtain the co-state equations

$$\nabla \times \mathbf{v} - \nabla q = \mathbf{T}_{\boldsymbol{\phi}}(\mathbf{x} + \mathbf{u}) \big(\mathbf{T}(\mathbf{x} + \mathbf{u}) - \mathbf{R}(\mathbf{x}) \big) \quad \text{in } \Omega, \tag{2.4}$$

where $\mathbf{T}_{\boldsymbol{\phi}}$ denotes the Jacobian of $\mathbf{T}(\boldsymbol{\phi})$.

Optimality conditions. The optimality conditions are obtained by setting the first variation of $\mathcal{L}(\mathbf{u}, f, \mathbf{g}, q, \mathbf{v})$ with respect to the controls f and $\mathbf{g}$ to

zero. For example, we have

$$\begin{aligned}\delta_f \mathcal{L} &= \lim_{\varepsilon\to 0}\frac{d}{d\varepsilon}\mathcal{L}(\mathbf{u}, f++\varepsilon f, \mathbf{g}, q, \mathbf{v})\\ &= \lim_{\varepsilon\to 0}\frac{d}{d\varepsilon}\Big\{\omega\int_\Omega f\delta f d\mathbf{x} - \int_\Omega q\big(\nabla\cdot\mathbf{u} - f - \varepsilon\delta f + 1\big)d\mathbf{x}\Big\}\\ &= \int_\Omega(\omega f - q)\delta f d\mathbf{x}\end{aligned}$$

and similarly for $\delta_{\mathbf{g}}\mathcal{L}$. Thus, setting $\delta_f\mathcal{L} = 0$ and $\delta_{\mathbf{g}}\mathcal{L} = \mathbf{0}$, we obtain the optimality conditions

$$f = \frac{1}{\omega}q \quad \text{and} \quad g_i = \frac{1}{w_i}v_i, \quad i = 1,2,3, \quad \text{in } \Omega. \tag{2.5}$$

Optimality system reformulated in terms of Poisson equations. From the optimality system (2.3)–(2.5), we easily obtain

$$\begin{cases}\Delta\mathbf{u} = \dfrac{1}{\omega}\nabla q - \dfrac{1}{w}\nabla\times\mathbf{v} & \text{in } \Omega\\ \mathbf{u} = \mathbf{0} & \text{on } \partial\Omega\end{cases} \tag{2.6}$$

$$\begin{cases}\Delta q = -\nabla\cdot\Big(\mathbf{T}_{\boldsymbol{\phi}}(\mathbf{x}+\mathbf{u})\big(\mathbf{T}(\mathbf{x}+\mathbf{u}) - \mathbf{R}(\mathbf{x})\big)\Big) & \text{in } \Omega\\ q = 0 & \text{on } \partial\Omega\end{cases} \tag{2.7}$$

$$\begin{cases}\Delta\mathbf{v} = -\nabla\times\Big(\mathbf{T}_{\boldsymbol{\phi}}(\mathbf{x}+\mathbf{u})\big(\mathbf{T}(\mathbf{x}+\mathbf{u}) - \mathbf{R}(\mathbf{x})\big)\Big) & \text{in } \Omega\\ \mathbf{v} = \mathbf{0} & \text{on } \partial\Omega,\end{cases} \tag{2.8}$$

where for (2.6) and (2.8) we have used the identity $\Delta\mathbf{v} = \nabla(\nabla\cdot\mathbf{v}) - \nabla\times(\nabla\times\mathbf{v})$, for (2.6) we have set $w_i = w$ for $i = 1,2,3$, and for (2.8), we have adopted the gauge $\nabla\cdot\mathbf{v} = 0$. Next we briefly study numerical implementation of the present method.

3. Numerical implementation

- Suppose we have control parameters.
- Obtain $\mathbf{u}$ from decoupled state equations (2.6).
- Obtain q, $\mathbf{v}$ from decoupled costate equations (2.7), (2.8).
- Next get new controls from the optimality conditions (2.5).
- Normalize controls and repeat the same process until the error condition is satisfied or a present number of iterations is achieved.

Although there are several sophisticated numerical methods to solve these Poisson equations, we use successive-over relaxation method in 3D case and finite-difference multigrid method in 2D case. Because these two methods are quite well-known in the solutions of the Poisson equations we skip their details here.

4. Noise

Noise constitutes an important limitation in the registration of images. Typically, noise is assumed to be Gaussian, Laplacian or Poisson distributed. We can separate the noisy image models as additive and multiplicative noise models. In the additive noise model observed image $u_0(x)$ is sum of the true image $u(x)$ and noise $\eta(x)$:

$$u_0(x) = u(x) + \eta(x).$$

In this model of additive noise, in general, the noise $\eta(x)$ is commonly modeled by Gaussian noise of mean m and variance v. A multiplicative noise is modeled as

$$u_0(x) = u(x)\eta(x).$$

In this case, $\eta(x)$ is uniformly distributed random noise with mean m and variance v.

5. Some computational examples

Example 1. (**Registration of the 3D noise-free images**) Let $\Omega = [0,1] \times [0,1] \times [0,1]$ and the following discrete images have a resolution of $16 \times 16 \times 16$ pixels. Let the reference image, $R(x,y,z)$ and the template image, $T(x,y,z)$ be given as follows:

$$R(x,y,z) = \begin{cases} 10, & \text{d(x,y,z)}\leq 0; \\ 9.5 + 5(0.1 + 1.5d(x,y,z)), & 0 \leq d(x,y,z) \leq 2; \\ 25, & 2 \leq d(x,y,z). \end{cases}$$

where

$$d(x,y,z) = \sqrt{(x-8)^2 + (y-8)^2 + (z-9)^2} - 3.$$

$$T(x,y,z) = \begin{cases} 10, & \text{d(x,y,z)}\leq 0; \\ 9.5 + 5(0.1 + 3d(x,y,z)), & 0 \leq d(x,y,z) \leq 1; \\ 25, & 1 \leq d(x,y,z). \end{cases}$$

where

$$d(x,y,z) = \sqrt{(x-8)^2 + (y-8)^2 + (z-9)^2} - 2,$$

with Dirichlet boundary conditions.

Figures 3 illustrates the slices of the registered image. Table 1 demonstrates the results of registration. Our method is quite fast and registers the given 3D images with a good quality.

Example 2. (**Registration in the presence of noise in the 3D case**) In this example we study the effect of noise on image registration. We register the transformed template image with the noisy images. We determine the approximate noise level at which our registration method fails. Initially, we

Table 1: SSD and Average duration of implementation

Iterations	SSD with SOR	SOR Duration
1	1674.6	1 sec
5	209.9	2 sec
60	16.5	25 sec
200	3.5	1 min

consider the registration problem in which template image is noise free but the reference image contains certain level of noise. We added increasing level of additive noises to the reference image described in the previous problem with the noise densities of 0.22, 0.44, 0.74. Table 2 demonstrates the change in the SSD versus noise. As we see from the table our method fails only when the multiplicative noise level is 0.74.

Table 2: SSD with additive noise and Average durations of implementations

Iterations	SSD-0.22	SSD-0.44	SSD-0.74	Average Duration
1	1893.5	2140.8	2900.5	1 sec
2	1342.9	1401.4	2906.1	3 sec
120	55.0	209.5	2900.1	1 min
200	47.4	160.4	2900.1	2 mins

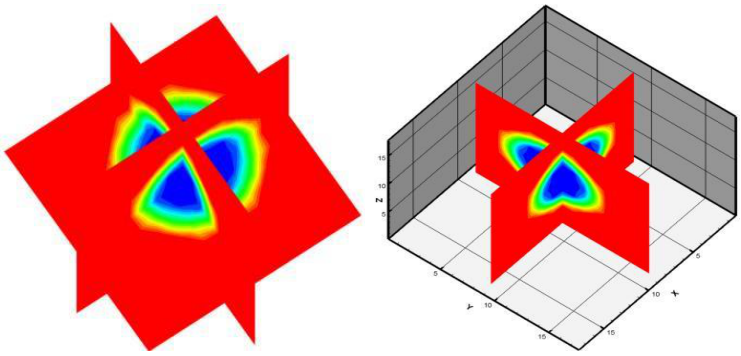

Figure 1: 3D and 2D slices

Example 3. (**Registration in the presence of noise in the 2D case**) In this example we define reference and template images as follows:

$$R(x,y) = \begin{cases} 10, & \text{d(x,y)}\leq 0; \\ 9.5 + 5(0.1 + 1.5d(x,y)), & 0 \leq d(x,y) \leq 2; \\ 25, & 2 \leq d(x,y). \end{cases}$$

where

$$d(x,y) = \sqrt{1.6(x-30)^2 + (y-30)^2} - 12.$$

$$T(x,y) = \begin{cases} 10, & \text{d(x,y)}\leq 0; \\ 9.5 + 5(0.1 + 3d(x,y)), & 0 \leq d(x,y) \leq 1; \\ 25, & 1 \leq d(x,y). \end{cases}$$

where

$$d(x,y) = \sqrt{(x-35)^2 + 1.5(y-35)^2} - 7.$$

We add certain level of noise to both reference and template images. In order to see the effect of the noise to the registration procedure, we multiplied both reference and template images described above for the 2D case with the noise densities of 0.22, 0.44, 0.66, 0.90. Figure 2 shows the noisy images resulting from multiplying the reference image with the above mentioned noise densities. Table 3 demonstrates the change in the SSD versus noise. As we see from the table our method fails when the multiplicative noise level is 0.90, which is quite good.

Table 3: SSD with additive noise and Average durations of implementations

Iterations	SSD-0.22	SSD-0.44	SSD-0.66	SSD-0.90	Aver.Dur.
1	2106.4	2334.9	2576.1	2774.7	1 sec
5	1627.5	1874.3	2001.8	2672.3	3 sec
20	908.3	1174.7	1874.7	2672.3	6 sec
120	14.2	25.7	173.7	2672.3	34 sec
200	13.1	16.7	164.1	2672.3	1 min

Notice that it is possible to make experiments for different situations: For example, adding or multiplying reference and template images with the same or different noise densities. Although we present the results of image registration experiments using only multiplicative and additive Gaussian noise in this paper, we have also conducted numerous experiments using other types of noise, including Salt-and-Pepper and Poisson distributed noises. Obtained results are similar and for the sake of the brevity we omit showing them here.

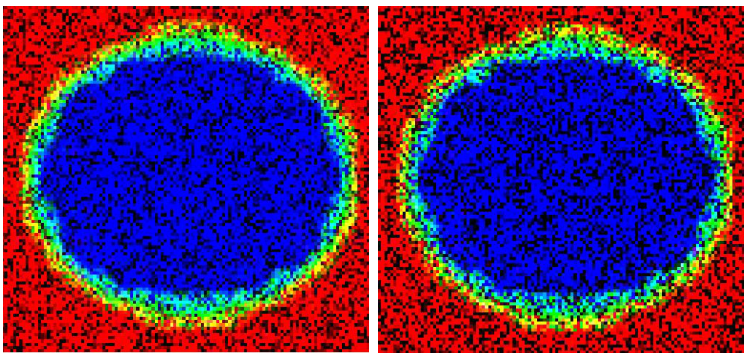

Figure 2: Noisy images with noise densities: 0.44, 0.90.

Example 4. In this example we register the template image to rotated noisy images shown in Figure 3. Since reference image is a rigid transformation of the template image (shown in the leftmost side of Figure 3), we will restrict the registration process to linear transformations which is easily taken care of by the curl term with our method. Notice that rotated images shown in Figure 3 were obtained by rotating the original image 35° counter clockwise. We added increasing level of additive Gaussian noises to the template image with the imaging densities of 0.10, 0.22, 0.44, 0.66, 0.90. Figure 3 shows the resulting noisy images. Table 4 indicates the change in the SSD versus noise. As we see from the table our method fails only when the additive Gaussian

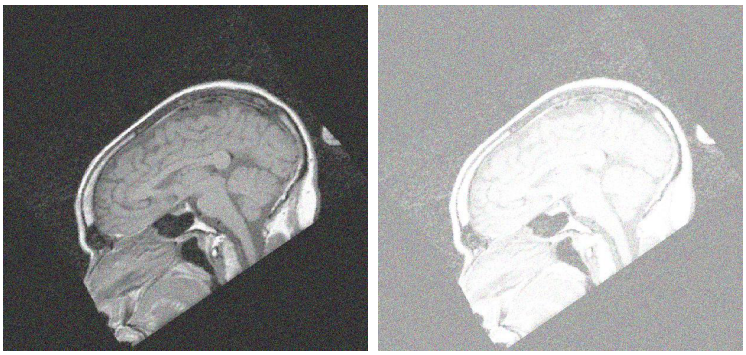

Figure 3: Rotated noisy images with noise densities: 0.22, 0.66.

noise level is 0.90. Finally let us emphasize that we dropped the boundary conditions in this case.

Table 4: SSD with additive Gaussian noise and Duration of implementation

Iterations	SSD-0.22	SSD-0.44	SSD-0.66	SSD-0.90	Aver.Dur.
1	1208.3	1955.8	2476.1	2774.7	1 sec
5	301.5	909.0	1430.4	2972.3	3 sec
120	5.4	12.5	27.0	2972.3	34 sec
200	5.1	11.9	23.3	2972.3	1 min

6. Summary and conclusion

In this article we have presented a systematic method for the nonrigid image registration. We applied our method to some 2D and 3D images which contain a certain level of noise. We implemented our algorithm in Fortran language on Intel Pentium D workstation running on the UNIX operating system. Computational examples given in the previous section were used to test the algorithm. We used discrete images which have a resolution of 64×64 pixels in the 2D case and $16 \times 16 \times 16$ in the 3D case. Image registration experiments demonstrate the accuracy and efficiency of the registration method. In a related future work we shall compare the results of our method with some other well-known methods such as elastic, fluid, diffusion and curvature-based techniques [1] for the registration of both noisy and noise-free images. The programming codes used in the implementation of the algorithms can be seen in [5].

References

[1] J. Modersitzki, *Numerical Methods for Image Registration*, Oxford University Press, 2004.
[2] M. A. Akinlar, *A New Method for Non-rigid Registration of 3D Images*, Ph.D. thesis, The University of Texas at Arlington, 2009.
[3] B. Fischer and J. Modersitzki, *Curvature Based Image Registration* J. Math. Imaging Vis., 18 (1) (2003), 81-85.
[4] E. Lee and M. Gunzburger, *An optimal control formulation of an image registration problem*, J. Math. Imaging Vis., 36 (1) (2010), 69-80.
[5] http://myweb.sabanciuniv.edu/akinlar/

(Received: May 26, 2010)
(Revised: September 20, 2010)

Mehmet Ali Akinlar
Department of Scientific Computing
Florida State University
Tallahassee, FL, 32306-4120, USA
E-mail: mehmetaliakinlar@gmail.com

Ranis N. Ibragimov
Department of Mathematics
The University of Texas at Brownsville
Brownsville, TX, 78520, USA and
Research and Support Center for
Applied Mathematical Modeling
New Mexico Inst. of Mining and Technology
Socorro, NM, 87801, USA.