A Super Fast Registration Algorithm

Bernd Fischer and Jan Modersitzki

Institute of Mathematics
Medical University of Lübeck, 23560 Lübeck
Email: {fischer,modersitzki}@math.mu-luebeck.de

Abstract. Image registration is an often encountered problem in digital imaging, in particular in medical imaging. In most applications simple rigid deformations are not satisfactory and complex, non-rigid and non-linear deformations must be employed. A large class of non-rigid, parameter-free, matching techniques minimizes the distance of the given images subject to a regularizing term. In this note we propose a novel scheme for automatic registration by introducing a specific regularizing term. We show that the complexity of its implementation is linear with respect to the size of the images and demonstrate its performance. Moreover, we draw a connection to Thirion's demon based approach.

1 Introduction

In the last decade a number of non-rigid, automatic registration algorithms have been proposed, see, for example, [1, 2, 3, 4, 5] and references therein. Most of these schemes may be viewed as a procedure which minimizes a suitable distance measure subject to a regularization term. There are essentially two approaches to solve these optimization problems numerically. One is to deal directly with the original formulation, whereas the other is to solve a related partial differential equation. Here, we focus on the latter method. Typical members out of this class are the elastic [4] and fluid [5] deformation models. The elastic model requires the repeated solution of the Navier-Lamé equation. In contrast, for the fluid model one is tempted to solve a simple version of the Navier-Stokes equation. It seems to be a conventional wisdom that a finite difference approximation to these equations leads to schemes which are by far to slow (see, e.g., [6]). The main bottleneck is thought of to be the solution of the corresponding linear systems. However, Fischer and Modersitzki [7] recently showed that one may solve these systems in just $\mathcal{O}(N \log N)$ operations, where N is the number of pixel.

In this note we propose a novel gradient based penalizing term and devise a super fast and stable implementation for a finite difference approximation of the underlying partial differential equation. Actually, we show that the solution of the corresponding linear system requires only $\mathcal{O}(N)$ operations, that is, its complexity is linear with respect to the number of pixel.

Beside this, we discuss Thirion's [8] demon based approach. He proposed a method which works well in practice but its derivation is guided by intuition and not entirely understood. In the literature there have been several attempts to shed some light on his approach (see, e.g., [9, 6]). Because Thirion offers a variety of possible implementations, the underlying theory is widespread. However, the bottom line is, that he calculates the deformations by regularizing certain driving forces by a Gaussian convolution filter. We show that this technique may be viewed as a special (low order) approximation to the partial differential equation connected to our new scheme and thereby gaining some insight into Thirion's work.

2 Approach

We refer to the template image as $T(\mathbf{x})$ and the study image as $S(\mathbf{x})$ where $\mathbf{x} \in \Omega = [0,1]^d$. The registration algorithm described in this paper is applicable to images with any number of dimensions d. For a particular point $\mathbf{x} \in \Omega$, the value $T(\mathbf{x})$ is the intensity at $\mathbf{x}$. The purpose of the registration is to determine a transformation, sometimes called warping, of $T(\mathbf{x})$ onto $S(\mathbf{x})$. Ideally, one wants to determine a displacement field $\mathbf{u} : \Omega \to \Omega$ such that $T(\mathbf{x}-\mathbf{u}(\mathbf{x})) = S(\mathbf{x})$. The question is how to find such a mapping $\mathbf{u}$. A straightforward approach would be to minimize the following distance measure

$$I(\mathbf{u}) = \frac{1}{2}\int_{\Omega} \left(T(\mathbf{x}-\mathbf{u}(\mathbf{x})) - S(\mathbf{x})\right)^2 d\mathbf{x}. \tag{1}$$

Of course other functionals, like the mutual information based measure, might be used as well. The theory is along the same lines, but will not be considered here. In order to rule out discontinuous and/or suboptimal solutions of the above minimization problem, or to privilege likely solutions it is common to introduce a smoothing or regularizing term $E(\mathbf{u})$. The problem now reads, find a mapping $\mathbf{u}$ which minimizes the joint criterion

$$J(\mathbf{u}) = \alpha E(\mathbf{u}) + I(\mathbf{u}). \tag{2}$$

For example, the well-known deformable grid methods based on elasticity or fluid mechanics may be phrased in terms of minimizing the functional (2) for specific choices of $E(\mathbf{u})$. We note that the parameter α determines the relative weight of the regularizing term. Since the issue of the choice of α is not addressed here, we set $\alpha = 1$. This is the parameter used in our experiments as well.

Let us now investigate the following stabilizer $E(\mathbf{u})$ which is designed to penalize oscillating deformations. For convenience, we formulate the problem explicitly for the three-dimensional case $d = 3$. Formulations for different dimensions are straightforward.

$$E(\mathbf{u}) = E(u_1,u_2,u_3) = \frac{1}{2}\int_{\Omega} \|\mathrm{grad}(u_1)\|_2^2 + \|\mathrm{grad}(u_2)\|_2^2 + \|\mathrm{grad}(u_3)\|_2^2 \, d\mathbf{x}. \tag{3}$$

In accordance with the calculus of variations, the function $\mathbf{u}$ which minimizes the functional (2) with respect to (3) has to satisfy the Euler-Lagrange equations

$$\begin{aligned} \Delta u_1(\mathbf{x}) &= (S(\mathbf{x}) - T(\mathbf{x}-\mathbf{u}(\mathbf{x}))\partial_1 T(\mathbf{x}-\mathbf{u}(\mathbf{x})) \\ \Delta u_2(\mathbf{x}) &= (S(\mathbf{x}) - T(\mathbf{x}-\mathbf{u}(\mathbf{x}))\partial_2 T(\mathbf{x}-\mathbf{u}(\mathbf{x})) \\ \Delta u_3(\mathbf{x}) &= (S(\mathbf{x}) - T(\mathbf{x}-\mathbf{u}(\mathbf{x}))\partial_3 T(\mathbf{x}-\mathbf{u}(\mathbf{x})) \end{aligned} \tag{4}$$

in Ω subject to appropriate boundary conditions. Here Δ denotes the Laplace operator. The so-called force field

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = (T(\mathbf{x} - \mathbf{u}(\mathbf{x})) - S(\mathbf{x}))\ \mathrm{grad}(T(\mathbf{x} - \mathbf{u}(\mathbf{x}))) \tag{5}$$

is used to drive the deformation. It is worth noticing that $\mathbf{f}$ is the derivative of the functional $I(\mathbf{u})$ with respect to $\mathbf{u}$. Changing I results in a different force field (see the comment after (1)).

A popular approach to solve a non-linear system of partial differential equations like (4) is to introduce an artificial time t and to compute the steady state solution $\partial_t \mathbf{u}(\mathbf{x}, t) = 0$ of the time dependent partial differential equation

$$\partial_t \mathbf{u}(\mathbf{x}, t) = \Delta \mathbf{u}(\mathbf{x}, t) + \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{6}$$

in Ω via a time marching algorithm. More precisely, to solve (6), we employ the following semi-implicit iterative scheme

$$\partial_t \mathbf{u}^{k+1}(\mathbf{x}, t) - \Delta \mathbf{u}^{k+1}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, \mathbf{u}^k), \quad k = 1, 2, \ldots, \tag{7}$$

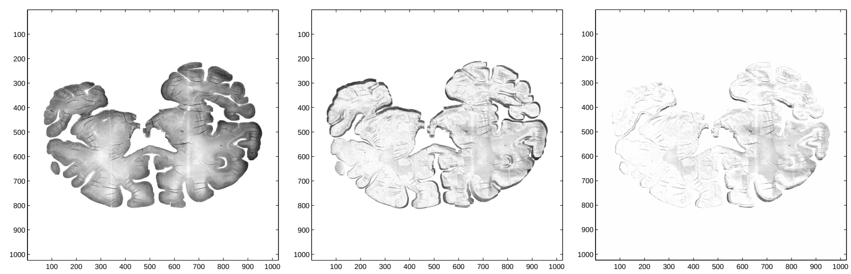
where $\mathbf{u}^0$ is some initial deformation, typically $\mathbf{u}^0 = 0$. In other words, the trick is to compute the driving force $\mathbf{f}$ for the previous solution $\mathbf{u}^k$ and subsequently to solve for $\mathbf{u}^{k+1}$.

There are several ways to solve (7). We start by noting that equation (7) is nothing but an inhomogeneous heat-equation and well understood (see, e.g., Folland [10]). Actually, if we would have to solve (7) with respect to the whole space $\Omega = \mathbb{R}^d$ then, under mild conditions on the driving force $\mathbf{f}$, it is possible to come up with an analytic solution. A representative result in this direction reads (see [10]): if $\mathbf{f} \in L^1$, then the convolution $\mathbf{u}^{k+1}(\mathbf{x}, t) = K_t(\mathbf{x}) * \mathbf{f}(\mathbf{x}, \mathbf{u}^k),\ t > 0$, is well defined almost everywhere and is a distribution solution of (7). It will be even a classical solution if $\mathbf{f} \in C^k,\ k > 1$. Here $K_t(\mathbf{x}) = (4\pi t)^{-d/2} \exp\left(-\|\mathbf{x}\|_2^2/(4t)\right)$ denotes the Gaussian kernel. Hence in order to solve (7) with respect to the bounded region $\Omega = [0,1]^d$ one may approximate the Gaussian kernel by a Gaussian filter of suitable length, that is, to compute at each time step $\mathbf{u}^{k+1} = K_\sigma * \mathbf{f}(\mathbf{u}^k)$ the force convolved with a Gaussian filter K_σ with characteristic width σ. This approach is what Thirion calls *Demons 1: a complete grid of demons* (see [8]). However, he gives no hint on how to choose the parameter σ for a given application. It turns out in practice, that a proper choice of this free parameter is a tricky business. Also, it is hard to analyze the complexity of the Gaussian filter based implementation, as it is directly connected to the choice of σ and the treatment of the boundaries of the images.

Let us now present an alternative way to solve (7). We treat it as a parabolic partial differential equation. This point of view has several advantages. There exists an immense knowledge on the numerical solution of partial differential equations which often goes in hand with fast and stable implementations. Moreover, it is straightforward to incorporate the boundaries of the images within the code. Various boundary conditions are possible and easy accessible. As a representative example we illustrate these points by considering a finite difference discretization of (7). It is worth noticing that standard arguments show that this approach results in a better approximation to (7) as the Gaussian filter based implementation. Typically one chooses the intrinsic discretization provided by the pixel as computational grid. Then one approximates the time and space derivatives in (7) by finite differences taking into account the chosen boundary conditions. This results in a system of linear equations $\mathcal{A}\mathbf{u}^{k+1} = \mathbf{f}^{\,k}$, where, for convenience, $\mathbf{u}^{k+1}$ and $\mathbf{f}^{\,k}$ denote vectors of suitable length. Consequently, the main work of this approach is the repeated solution of this linear equation. A close inspection of the coefficient matrix $\mathcal{A}$ shows that it has a rich structure. Actually it is a block diagonal matrix with d identical blocks A, where the size of A corresponds to the number of pixel N. Consequently, the solution of the large system decouples into the solution of d systems with coefficient matrix A. Moreover and most important, it is possible to solve these systems directly by means of the so-called AOS-scheme which has a linear computational complexity $\mathcal{O}(N)$ (see, Weickert [11]). Moreover, the implementation in a parallel environment is straightforward.

3 Experiments

To illustrate the performance of the new approach based on the finite difference formulation we present the registration of two consecutive frontal sections from a series of histological tissue sections of a human brain (see [12] for further details). We are indebted to Dr. Oliver Schmitt (Institute of Anatomy, Medical University of Lübeck) for providing the medical data. Fig. 1 displays the arbitrarily chosen section 3799 of size 1024×1024 pixel and the difference to section 3800 before and after registration. Note that the difference has been reduced by about 32%. Table 1 shows the performance on a SGI OCTANE (175 MHz, MIPS R10000, 128 MB RAM under IRIX 6.5) using MATLAB 5.3. In accordance with our theory, the CPU-times resemble nicely the linear behavior of the proposed scheme. It is expected to improve the execution time by applying a multi-resolution approach and by writing a proper C-implementation.



section 3799 difference$_{\text{before}} = 100\%$ difference$_{\text{after}} = 68\%$

Fig. 1. Histological frontal section of a human brain (left), difference to the following section before (middle) and after (right) registration.

Table 1. Execution time and floating point operations per pixel (flops/pixel) for one time-step and for different image sizes.

images	128^2	256^2	512^2	1024^2
cpu time	0.6s	2.6s	9.7s	36.7s
flops/pixel	72.3	70.1	59.8	50.3

4 Conclusion

We have indicated that a large class of registration schemes may be phrased in terms of a variational problem. The minimizer of such a problem is provided by the solution of an associated partial differential equation. This point of view offers a variety of well-developed and efficient algorithms. Along this lines we have developed a new registration algorithm for multiple dimensions and analyzed its computational complexity which happens to be linear.

Beside this, we showed that Thirion's demon based approach turns out to be a special implementation of an inhomogeneous heat-equation which constitutes a necessary condition for the variational problem considered here.

References

1. Broit C: Optimal registration of deformed images. PhD thesis, Computer and Information Science, University of Pensylvania, 1981.
2. Bajcsy R, Kovačič S: Multiresolution elastic matching. Computer Vision, Graphics and Image processing, 46:1–21, 1989.
3. Amit Y: A nonlinear variational problem for image matching. SIAM J. Sci. Comp., 15(1):207–224, 1994.
4. Christensen GE: Deformable shape models for anatomy. PhD thesis, Sever Institute of Technology, Washington University, 1994.
5. Bro-Nielsen M: Medical image registration and surgery simulation. PhD thesis, IMM, Technical University of Denmark, 1996.
6. Bro-Nielsen M, Gramkow C: Fast fluid registration of medical images. Visualization in Biomedical Computing (VBC'96), Lecture Notes in Computer Science, Vol. 1131, Springer, 267–276, 1996.
7. Fischer B, Modersitzki J: Fast inversion of matrices arising in image processing. Numerical Algorithms, 22:1–11, 1999.
8. Thirion JP: Image matching as a diffusion process: an analogy with Maxwell's demons. Medical Image Analysis, 2(3):243–260, 1998.
9. Pennec X, Cachier P, Ayache N: Understanding the "demon's algorithm": 3d non-rigid registration by gradient descent. 2nd Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI'99), 597–605, 1999.
10. Folland GB: Introduction to partial differential equations. Princeton University Press, Princeton, New Jersey, 2nd edition, 1995.
11. Weickert J: Anisotropic diffusion in image processing. Teubner, Stuttgart, 1998.
12. Modersitzki J, Schmitt O, Fischer B: Effiziente, nicht-lineare Registrierung eines histologischen Serienschnittes durch das menschliche Gehirn. Beitrag auf der BVM 2001.