Fast Fluid Registration of Medical Images

Morten Bro-Nielsen[1,2] and Claus Gramkow[1]

[1] Dept. of Mathematical Modelling
Technical University of Denmark, Bldg. 321
DK-2800 Lyngby, Denmark
[2] 3D-Lab, School of Dentistry, Univ. of Copenhagen,
Nørre Alle 20, DK-2200 Copenhagen N, Denmark
e-mail: bro@imm.dtu.dk WWW http://www.imm.dtu.dk/˜bro

Abstract. This paper offers a new fast algorithm for non-rigid Viscous Fluid Registration of medical images that is at least an order of magnitude faster than the previous method by Christensen et al. [4]. The core algorithm in the fluid registration method is based on a linear elastic deformation of the velocity field of the fluid. Using the linearity of this deformation we derive a convolution filter which we use in a scale-space framework. We also demonstrate that the 'demon'-based registration method of Thirion [13] can be seen as an approximation to the fluid registration method and point to possible problems.

1 Introduction

Non-rigid registration of two medical images is performed by applying global and/or local transformations to one of the images (which we will call the template T) in such a way that it matches the other image (the study S). It is important to understand that the aim of the transformation is to map the template *completely* onto the study in such a way that information from the template can be applied to the study as well. A very important application of non-rigid registration is using an electronic atlas to segment a study image. Only if the transformation maps the template completely onto the study can the atlas be used to infer conclusions about the contents of the study. In practice this means that the transformation must accommodate both very complex and large deformations.

Bajcsy et al. [1, 10] were the first to demonstrate volumetric non-rigid registration of medical images. Building on initial work by Broit [2], they modelled the template image as a linear elastic solid and deformed it using forces derived from an approximation of the local gradient of a correlation based similarity measure. Multi-resolution were used to increase the speed.

Evans et al. [8] used anatomical landmarks to drive the deformation, and deformed the solid using a globally elastic model. Miller, Christensen et al. [3, 4] also used a globally elastic model, but derived the driving force from the derivative of a Gaussian sensor model.

These previous approaches to non-rigid registration have all suffered from the use of either *global* transformations or *small deformation* assumptions (as used in linear elasticity).

In [5] Christensen et al. extended their work and described a registration approach in which they use a viscous fluid model to control the deformation. The template image is modelled as a thick fluid that flows out to match the study under the control of the same derivative of a Gaussian sensor model they used in [4]. In [5] Christensen argue that this gaussian sensor model theoretically is better than the correlation based similarity measure used by Bajcsy et al. [1].

Elastic models constrain the possible deformation because the deformation is a compromise between internal and external forces. Elastic displacements do not reach the desired deformation because of internal strain in the elastic continuum. In a viscous fluid model, internal forces disappear over time and the desired deformation can be fully achieved.

Consequently, the fluid registration method satisfies the general requirements of both complex and large deformations and we therefore regard this method as the most advanced registration method available.

Unfortunately, the algorithm proposed by Christensen et al. is rather slow. They originally implemented the algorithm using a massively parallel DECmpp 128x64 MasPar computer on which the algorithm used on the order of 5-10 minutes for 2D and 2-6 hours for 3D registrations. In a recent paper [6] they show estimates of the execution time on a MIPS R4400 processor on the order of 2 hours for 2D and 7 days for 3D. In practice this means that the algorithm is not feasible unless a massively parallel computer is available.

The contribution of this paper is a new fast algorithm based entirely on convolution with filters which gives a *speed-up of at least an order of magnitude.*

2 Theory

In this section we describe our new algorithm for solving the viscous fluid registration problem. Without loss of generality the theory is described in the 2D case, but it is readily extendable to the 3D case.

In the first section we describe the original viscous fluid algorithm by Christensen et al. [5]. We define the template and study images and their relationship. The viscous fluid model is introduced along with the driving force and the numerical solution method. In the second section we discuss the core part of Christensen's numerical solution and introduce the general idea behind the convolution approach that we propose to increase the speed of the method. In the third section, the basic filter for the convolution approach is finally derived.

2.1 Fluid registration

We define the template image as $T(\boldsymbol{x})$ and the study image as $S(\boldsymbol{x})$ where $\boldsymbol{x} \in [0;1]^2$. The purpose of the registration is to determine a warping of $T(\boldsymbol{x})$ onto $S(\boldsymbol{x})$.

Eulerian reference frame In elastic deformation, particles are usually tracked by their initial coordinates, ie. the parametrization of the object. This sort of reference frame is called *Lagrangian*. But in fluid deformation the Lagrangian reference frame is inefficient and an *Eulerian* reference frame is used instead. In the Eulerian reference frame the particles are tracked based on their current/final position. Consequently, a particle at position $\boldsymbol{x} = [x_1, x_2]^T$ in the template image at time t originated at position $\boldsymbol{t}(\boldsymbol{x}, t) = \boldsymbol{x} - \boldsymbol{u}(\boldsymbol{x}, t)$ at time t_0 $(t > t_0)$, where $\boldsymbol{u}$ is the displacement. Notice that in the Eulerian reference frame, $\boldsymbol{u}(\boldsymbol{x}, t)$ describes the displacement of the particles as they move through $\boldsymbol{x}$. The Eulerian velocity field is determined by:

$$\boldsymbol{v}(\boldsymbol{x}, t) = \delta\boldsymbol{u}(\boldsymbol{x}, t)/\delta t + \nabla\boldsymbol{u}(\boldsymbol{x}, t)\boldsymbol{v}(\boldsymbol{x}, t) \tag{1}$$

where ∇ is the gradient operator. The term $\nabla\boldsymbol{u}(\boldsymbol{x}, t)\boldsymbol{v}(\boldsymbol{x}, t)$ results from the chain rule of differentiation and accounts for the kinematic non-linearities of the particles.

Viscous fluid model In the Eulerian framework we can write the partial differential equation (PDE) for the viscous fluid deformation of the template as [5]:

$$\mu\varDelta\boldsymbol{v}(\boldsymbol{x}) + (\mu + \lambda)\nabla(\nabla \cdot \boldsymbol{v}(\boldsymbol{x})) = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) \tag{2}$$

where $\varDelta = \nabla^T\nabla$ is the Laplacian operator and $\nabla(\nabla \cdot \boldsymbol{v})$ is the divergence operator. The force field $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}))$ is used to drive the flow. Those familiar with elasticity theory will recognize that for constant force $\boldsymbol{f}$ this is actually the PDE for linear elasticity working on the velocity field $\boldsymbol{v}$. The equation, therefore, works by elasticly smoothing the instantaneous velocity field of the fluid.

The term $\varDelta\boldsymbol{v}$ is also called the viscous term because it constrains the velocity field spatially. The $\nabla(\nabla \cdot \boldsymbol{v})$ term allows for contraction or expansion of the fluid.

The force field is defined as the derivative of a cost function C. For MRI images a Gaussian sensor model appears to be an appropriate model of the variation between the template and study image [5, 11]. The cost function and its derivative are:

$$C(T(\boldsymbol{x}), S(\boldsymbol{x}), \boldsymbol{u}) = \frac{1}{2}\int_{\Omega} \mid T(\boldsymbol{x} - \boldsymbol{u}(\boldsymbol{x}, t)) - S(\boldsymbol{x}) \mid^2 d\boldsymbol{x} \tag{3}$$

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}, t)) = -[T(\boldsymbol{x} - \boldsymbol{u}(\boldsymbol{x}, t)) - S(\boldsymbol{x})]\nabla T|_{\boldsymbol{x}-\boldsymbol{u}(\boldsymbol{x}, t)} \tag{4}$$

Numerical solution Solution of the viscous fluid registration problem requires solving the PDE [5]:

$$\mu\varDelta\boldsymbol{v}(\boldsymbol{x}, t) + (\mu + \lambda)\nabla(\nabla \cdot \boldsymbol{v}(\boldsymbol{x}, t)) = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}, t)) \tag{5}$$

$$\frac{\delta\boldsymbol{u}(\boldsymbol{x}, t)}{\delta t} = \boldsymbol{v}(\boldsymbol{x}, t) - \nabla\boldsymbol{u}(\boldsymbol{x}, t)\boldsymbol{v}(\boldsymbol{x}, t) \tag{6}$$

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}, t)) = -[T(\boldsymbol{x} - \boldsymbol{u}(\boldsymbol{x}, t)) - S(\boldsymbol{x})]\nabla T|_{\boldsymbol{x}-\boldsymbol{u}(\boldsymbol{x}, t)} \tag{7}$$

which includes non-linearities in both the force and the material derivative. To solve this problem we apply Euler integration over time using a forward finite difference estimate of the time derivative in equation 6:

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x},t_{i+1}) &= \boldsymbol{u}(\boldsymbol{x},t_i) + (t_{i+1}-t_i)(\boldsymbol{I} - \nabla \boldsymbol{u}(\boldsymbol{x},t_i))\boldsymbol{v}(\boldsymbol{x},t_i) \\ &= \boldsymbol{u}(\boldsymbol{x},t_i) + (t_{i+1}-t_i)\nabla \boldsymbol{t}(\boldsymbol{x},t_i)\boldsymbol{v}(\boldsymbol{x},t_i) \end{aligned} \tag{8}$$

Reliable Euler integration requires a well-conditioned transformation gradient $\nabla \boldsymbol{t}(\boldsymbol{x},t_i)$. Since the Jacobian $J = \mid \nabla \boldsymbol{t}(\boldsymbol{x},t_i) \mid$ provides a measure of the condition of $\nabla \boldsymbol{t}(\boldsymbol{x},t_i)$ we require $J > 0$.

The transformation becomes singular for large curved transformations because of the discretization. To evade this problem we apply the same regridding method as Christensen [5]. Every time the Jacobian J drops below 0.5 we generate a new template by applying the current deformation. In addition the displacement field is set to zero, whereas the current velocities remain constant. The total deformation becomes the concatenation of the displacement fields associated with the sequence of propagated templates.

The complete algorithm for solving the viscous fluid registration problem consequently becomes [5]:

1. Let $i = 0$ and $\boldsymbol{u}(\boldsymbol{x},0) = 0$
2. Calculate the body force $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}(\boldsymbol{x},t_i))$ using equation 7.
3. If $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}(\boldsymbol{x},t_i))$ is below a threshold for all $\boldsymbol{x}$, then STOP.
4. Solve the linear PDE equation 5 for instantaneous velocity $\boldsymbol{v}(\boldsymbol{x},t_i)$ and force $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}(\boldsymbol{x},t_i))$.
5. Choose a timestep $(t_{i+1}-t_i)$ so that $\nabla \boldsymbol{t}(\boldsymbol{x},t_i)\boldsymbol{v}(\boldsymbol{x},t_i) < d\boldsymbol{u}_{max}$, where $d\boldsymbol{u}_{max}$ is the maximal flow allowed in one iteration (0.7 in this work).
6. Perform Euler integration using equation 8.
7. If the Jacobian $J = \mid \nabla \boldsymbol{t}(\boldsymbol{x},t_i) \mid$ is less than 0.5 then regrid the template.
8. $i = i + 1$, goto 1

The only remaining question is how to solve the PDE equation in step 4. We discuss this in the following section.

2.2 Solving the linear PDE

In the algorithm shown above, the core problem is solving the linear PDE:

$$\boldsymbol{\mathcal{L}}\boldsymbol{v} = \mu \Delta \boldsymbol{v} + (\mu + \lambda)\nabla(\nabla \cdot \boldsymbol{v}) = \boldsymbol{f} \tag{9}$$

for constant force and time. In practice, solving this PDE is the time consuming part of the fluid registration. The contribution of the rest of the paper is a fast way of doing this.

As we saw previously, for constant force $\boldsymbol{f}$ and time t this PDE is linear and the linear operator $\boldsymbol{\mathcal{L}}$ is the linear elasticity operator working on $\boldsymbol{v}$. Linear elastic problems are normally solved using implicit finite element or finite difference methods. But in the case of images, we assign nodes in the elastic model to each pixel or voxel. The size of the problem, therefore, is huge and in practice unsolvable with these methods. Instead explicit methods must be used.

Christensen et al. [5] use successive overrelaxation (SOR) with checker board update to solve the linear elastic problem.

We suggest solving the the linear PDE using scale-space convolution. Using the linearity of the PDE and the superposition principle, we create a filter as the impulse response of the linear operator $\boldsymbol{\mathcal{L}}$ and subsequently apply this filter to the force field $\boldsymbol{f}$.

This work has been inspired by the work of Nielsen et al. [12], who show that Tikhonov regularization can be implemented using Gaussian scale-space, and Thirion [13], who propose a 'demon'-based registration algorithm which we will show later, is an approximation to the viscous fluid registration problem.

2.3 Convolution filter for linear elasticity

In this section we develop the convolution filter used to solve the linear PDE. First the displacement field $\boldsymbol{v}$ is decomposed using the eigen-function basis of the linear operator $\boldsymbol{\mathcal{L}}$. Then the impulse response of the linear operator is determined in this basis. We note that the impulse response of a linear operator is a filter that implements the operator. Finally, we discretize the impulse response to get a discrete filter.

Eigen-functions of the linear operator $\boldsymbol{\mathcal{L}}$ The eigen-functions of the linear operator $\boldsymbol{\mathcal{L}}$ using sliding boundary conditions are given by Miller, Christensen et al. in [3, 5] as:

$$\boldsymbol{\phi}_{ij1}(\boldsymbol{x}) = \alpha_1 \begin{bmatrix} ip(\boldsymbol{x}) \\ jq(\boldsymbol{x}) \end{bmatrix} \qquad \boldsymbol{\phi}_{ij2}(\boldsymbol{x}) = \alpha_2 \begin{bmatrix} -jp(\boldsymbol{x}) \\ iq(\boldsymbol{x}) \end{bmatrix} \tag{10}$$

with the eigen-values:

$$\kappa_{ij1} = -\pi^2(2\mu + \lambda)(i^2 + j^2) \qquad \kappa_{ij2} = -\pi^2\mu(i^2 + j^2) \tag{11}$$

where

$$p(\boldsymbol{x}) = \sin i\pi x_1 \cos j\pi x_2 \qquad q(\boldsymbol{x}) = \cos i\pi x_1 \sin j\pi x_2 \tag{12}$$

$$\alpha_1 = \alpha_2 = \sqrt{\frac{4}{\Gamma_{ij}(i^2 + j^2)}} \tag{13}$$

where

$$\Gamma_{ij} = \begin{cases} 1 & \mathit{if\ none\ of\ i,j\ are\ zero} \\ 2 & \mathit{if\ one\ of\ i,j\ is\ zero} \end{cases} \tag{14}$$

Using the new orthonormal basis, the velocity $\boldsymbol{v}$ can be decomposed into:

$$\boldsymbol{v}_N(\boldsymbol{x}) = \sum_{ij=0}^{N} \sum_{r=1}^{2} a_{ijr} \boldsymbol{\phi}_{ijr}(\boldsymbol{x}) \tag{15}$$

where a_{ijr} are the coefficients of the decomposition. N determines the number of basis functions included in the decomposition. Note that $\boldsymbol{v}(\boldsymbol{x}) = \lim_{N\to\infty} \boldsymbol{v}_N(\boldsymbol{x})$.

Determining the impulse response of $\boldsymbol{\mathcal{L}}$ We will now determine the impulse response of the linear operator $\boldsymbol{\mathcal{L}}$ for an impulse force $\hat{\boldsymbol{f}}$ in the x_1 direction at $\boldsymbol{c} = [0.5, 0.5]^T$. First the linear operator is applied to the decompostion of $\boldsymbol{v}$:

$$\begin{aligned}\hat{\boldsymbol{f}} &= \boldsymbol{\mathcal{L}}\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{\mathcal{L}}\sum_{ijr} a_{ijr}\boldsymbol{\phi}_{ijr}(\boldsymbol{x}) = \sum_{ijr} a_{ijr}\boldsymbol{\mathcal{L}}\boldsymbol{\phi}_{ijr}(\boldsymbol{x}) \\ &= \sum_{ijr} a_{ijr}\kappa_{ijr}\boldsymbol{\phi}_{ijr}(\boldsymbol{x}) \qquad (16)\end{aligned}$$

We then take the inner product $< \boldsymbol{a}, \boldsymbol{b} >= \int_\Omega \boldsymbol{a}^T\boldsymbol{b}d\boldsymbol{x}$ of the equation with $\boldsymbol{\phi}_{lms}(\boldsymbol{x})$:

$$\begin{aligned}&\sum_{ijr} a_{ijr}\kappa_{ijr} < \boldsymbol{\phi}_{ijr}(\boldsymbol{x}), \boldsymbol{\phi}_{lms}(\boldsymbol{x}) > = < \hat{\boldsymbol{f}}, \boldsymbol{\phi}_{lms}(\boldsymbol{x}) > \\ &\Updownarrow \\ &a_{lms}\kappa_{lms} = < \hat{\boldsymbol{f}}, \boldsymbol{\phi}_{lms}(\boldsymbol{x}) > \\ &\Updownarrow \\ &a_{lms} = \frac{1}{\kappa_{lms}} < \hat{\boldsymbol{f}}, \boldsymbol{\phi}_{lms}(\boldsymbol{x}) > = \frac{1}{\kappa_{lms}}\phi^{x_1}_{lms}(\boldsymbol{c}) \qquad (17)\end{aligned}$$

where $\phi^{x_1}_{lms}$ is the x_1-coordinate of $\boldsymbol{\phi}_{lms}$. In the step from line 1 to line 2, we used the fact that $< \boldsymbol{\phi}_{ijr}, \boldsymbol{\phi}_{lms} >$ is zero for $(i, j, r) \neq (l, m, s)$.

We can now write the decomposition of $\boldsymbol{v}(\boldsymbol{x})$ for the case of an impulse force applied in $\boldsymbol{c}$ as:

$$\begin{aligned}\boldsymbol{v}(\boldsymbol{x}) &= \sum_{ij}^{\infty}\sum_{r=1}^{2} a_{ijr}\boldsymbol{\phi}_{ijr}(\boldsymbol{x}) = \sum_{ij}^{\infty}\sum_{r=1}^{2}\frac{1}{\kappa_{lms}}\phi^{x_1}_{lms}(\boldsymbol{c})\boldsymbol{\phi}_{ijr}(\boldsymbol{x}) \\ &= \frac{4}{\pi^2\mu(2\mu+\lambda)}\sum_{ij}^{\infty}\frac{p(\boldsymbol{c})}{(i^2+j^2)^2\mathit{\Gamma}_{ij}}\begin{bmatrix}-(i^2\mu+(2\mu+\lambda)j^2)p(\boldsymbol{x}) \\ (\mu+\lambda)ijq(\boldsymbol{x})\end{bmatrix} \qquad (18)\end{aligned}$$

This gives us the impulse response of the linear operator $\boldsymbol{\mathcal{L}}$ for an impulse force in the x_1 direction applied in $\boldsymbol{c}$. The impulse response for the x_2 direction is determined by simple rotation of the response for the x_1 direction. In the next section we will see how this impulse response can be used to determine a discrete filter implementing the linear operator $\boldsymbol{\mathcal{L}}$.

Discretizing the impulse response In general the impulse response of the linear operator is the linear filter implementing the operator. But in the continuous case a force applied to a single point yields an infinitely large displacement of this particular point. However, in the discrete case we sample the filter on a discrete grid and apply a lowpass filtering with a cut-off at the Nyquist frequency to eliminate aliasing from higher order frequency components. The force is thereby smoothed over a small area or volume.

We note that the decomposition of the impulse response based on the eigenfunction basis is a frequency based decomposition. Big i and j correspond to high frequencies and small to low frequencies. We can therefore perform an ideal lowpass filtering of the impulse response by truncating the sequence at N instead of summing to infinity.

The sampled filter is defined with dimensions $D \times D$, D odd, in the domain $[0;1]^2$. The sampling interval is consequently $\theta = 1/(D-1)$ which Shannons sampling theorem relates to the cut-off frequency f by $\theta \leq 1/2f$. From equation 18 the frequencies corresponding to the summation variables are determined:

$$f_i = \frac{1}{2}i \qquad f_j = \frac{1}{2}j \tag{19}$$

and the common truncation point becomes $i = j = N = D - 1$. We can now collect everything in:

Theorem. *Consider a linear filter of size $D \times D$, and let the lattice be addressed by $\boldsymbol{y} = [y_1, y_2]^T$, where $y_r \in [-\frac{D-1}{2}, \frac{D-1}{2}] \cap N$, $r = 1, 2$. The filter implementing the linear elastic operator $\boldsymbol{\mathcal{L}}$ for the x_1 direction is then:*

$$\boldsymbol{v}(\boldsymbol{x}) = \frac{4}{\pi^2\mu(2\mu+\lambda)} \sum_{ij=0}^{D-1} \frac{p(\boldsymbol{c})}{(i^2+j^2)^2\mathit{\Gamma})ij} \begin{bmatrix} -(i^2\mu + (2\mu+\lambda)j^2)p(\boldsymbol{x}) \\ (\mu+\lambda)ijq(\boldsymbol{x}) \end{bmatrix} \tag{20}$$

where

$$\boldsymbol{x} = \frac{1}{D-1}\boldsymbol{y} + \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} \tag{21}$$

□

We leave it to the reader to find the filter component for the x_2 direction.

To show that the filter actually works, we have made some experiments using the filter as the linear elasticity operator and comparing the results with a Finite Element (FEM) implementation of linear elasticity. The results have shown quite similar deformations.

2.4 Summary

In the previous sections we have described the original theory of the viscous fluid registration method and developed a convolution filter for the linear operator used in the core routine of the fluid registration.

Because of the limited span of the filter, we have implemented the viscous fluid registration algorithm using the filter in scale-space. The fluid registration is first performed on a rough scale. The result of this scale is then propagated to a finer scale and the fluid registration restarted here. This process is continued down to the finest scale of the scale-space, yielding the final registration result.

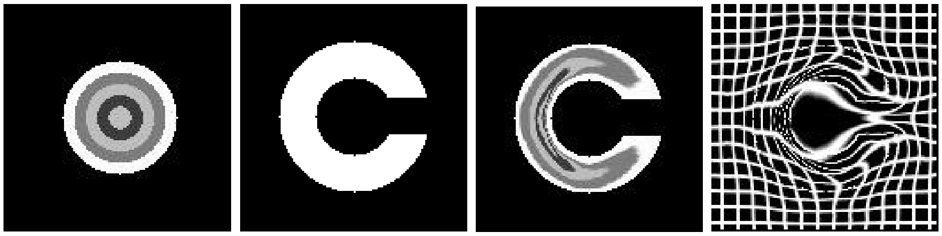

Fig. 1. Circle deforming into a 'C' using viscous fluid registration. From left to right: 1. Template. 2. Study. 3. Deformed template. 4. Grid showing the deformation applied to template.

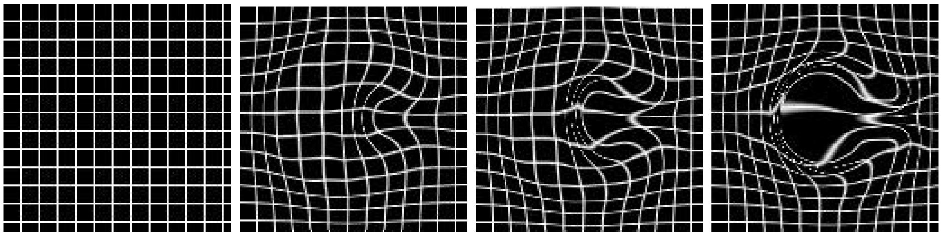

Fig. 2. Development of deformation of grid for viscous fluid registration of circle into 'C'.

3 Results

Figure 1 shows our results of registering a circle to a 'C' using viscous fluid deformation. The grid shows the curved and very large deformations that are applied to the template. Although the deformation is very large, the topology of the template is maintained. This is very important because it ensures that topology is maintained. Figure 2 shows the developing deformation as the fluid circle deforms into the 'C'. These results are very similar to figures 10.20-23 in [5]. Figure 3 show the result for two adjacent CT slices.

In general, our results are very good and similar to those of Christensen et al. But our timings are quite different. We have achieved stable timings on a *single processor* workstation similar to those stated by Christensen et al. for computations on a 128x64 DECmpp 12000 Sx/Model 200 massively parallel computer. When compared to estimates of timings for a MIPS R4400 processor [6] we can conclude that we achieve a speed-up of at least an order of magnitude. We hope to be able to share the data used by Christensen et al. for more elaborate comparison.

3.1 Comparison with 'demon'-based registration

In [13] Thirion proposed a 'demon'-based registration method. This is an iterative algorithm, where forces are determined in the template image based on

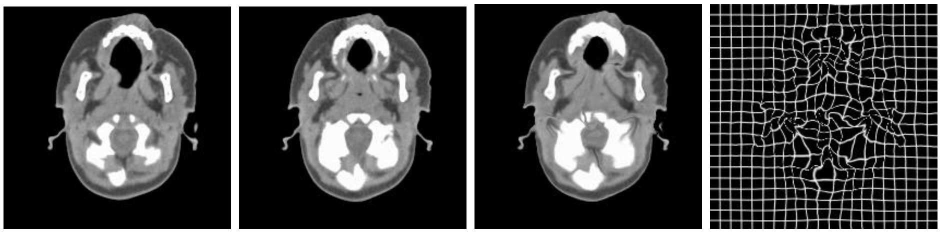

Fig. 3. CT slice registered to another slice using viscous fluid model. From left to right: 1. Template. 2. Study. 3. Deformed template. 4. Grid showing the deformation applied to template.

equations that are very similar to the body force used in this work. His equation 4:

$$\boldsymbol{f} = \frac{-(T-S)\nabla T}{\nabla T^2 + (T-S)^2} \tag{22}$$

is in fact just a normalized version of the body force used here (equation 7).

When Thirion deforms an image to match another, he performs an iterative process where body forces are determined using equation 22, the force field is lowpass filtered using a Gaussian filter and finally integrated over time.

Comparing the 'demon'-based algorithm with the algorithm in this paper, we see that:

- The body forces are almost the same.
- The lowpass filtering using a Gaussian corresponds to our application of the linear elastic filter.
- The time integration of the lowpass filtered force field corresponds to the Euler integration performed using equation 6.

We therefore conclude that the approach proposed in [13] is similar to the viscous fluid registration using convolution which we propose here. It is based on heuristics. Applying the Gaussian filter instead of the real linear elastic filter, is an approximation of the fluid model which could give problems in terms of topology and the stability of the fluid model.

4 Conclusion

In this paper we have shown that it is possible to speed-up the viscous fluid registration algorithm by Christensen et al. [5] by at least an order of magnitude. We achieve this speed-up by implementing the core process of the registration method using convolution with a filter we have developed. Our results are similar to those of Christensen et al.

The speed-up that we show here is based on a software implementation of the convolution. Use of specialized convolution hardware, as found in eg. the

RealityEngine II graphics board from SiliconGraphics, should speed-up the registration even more.

We have also showed that the 'demon'-based registration method by Thirion [13] is similar to the viscous fluid registration method developed here. This insight comes from the implementation of the core routine as a filter. Using this filter, the numerical implementations of the two methods look very similar. Since Thirion's method is a simplification, we suggest that it might have trouble handling more complex registration tasks.

Although we have only showed 2D results, the extension of the filter to the 3D case is straightforward, and we expect to do this in the future.

References

1. R. Bajcsy and S. Kovacic, *Multiresolution elastic matching*, Computer Vision, Graphics and Image Processing, 46:1-21, 1989
2. C. Broit, *Optimal registration of deformed images*, Doctoral dissertation, University of Pennsylvania, August 1981
3. M.I. Miller, G.E. Christensen, Y. Amit and U. Grenander, *Mathematical textbok of deformable neuroanatomies*, Proc. Natl. Acad. Sci. USA, 90:11944-11948, December 1993
4. G.E. Christensen, M.I. Miller and M. Vannier, *A 3D deformable magnetic resonance textbook based on elasticity,* in AAAI Spring Symposium Series: Applications of Computer Vision in Medical Image Processing, pp. 153-156, Stanford University, March 1994
5. G.E. Christensen, *Deformable shape models for anatomy*, Washington University Ph.D. thesis, August 1994
6. G.E. Christensen, M.I. Miller, M. Vannier and U. Grenander, *Individualizing neuroanatomical atlases using a massively parallel computer*, IEEE Computer, 29(1):32-38, January 1996
7. D.L. Collins, T.M. Peters, W. Dai, A.C. Evans, *Model-based segmentation of individual brain structures from MRI data*, Proc. SPIE Visualization in Biomedical Computing (1808), pp. 10-23, 1992
8. A.C. Evans, W. Dai, L. Collins, P. Neelin, S. Marret, *Warping of a computerized 3-D atlas to match brain image volumes for quantitative neuroanatomical and functional analysis*, Proc. SPIE Medical Imaging V (1445), pp. 236-246, 1991
9. K.H. Höhne, M. Bomans, M. Riemer,R. Schubert, U. Tiede and W. Lierse, *A 3D anatomical atlas based on a volume model*, IEEE Computer Graphics Applications, 12(4):72-78, 1992
10. J.C. Gee, M. Reivich and R. Bajcsy, *Elastically deforming atlas to match anatomical brain images*, Journal of Computer Assisted Tomography, 17(2):225-236, March 1993
11. E.R. McVeigh, R.M. Henkelman and M.J. Bronskill, *Noise and filtration in magnetic resonance imaging*, Medical Physics, 12(5):586-591, 1985
12. M. Nielsen, L. Florack and R. Deriche, *Regularization and Scale Space*, INRIA Tech. Rep. RR-2352, September 1994
13. J.-P. Thirion, *Non-rigid matching using demons*, Proc. Int. Conf. Computer Vision and Pattern Recognition (CVPR'96), 1996

This article was processed using the LaTEX macro package with LLNCS style