



HAL
open science

A Fast Nonrigid Image Registration With Constraints on the Jacobian Using Large Scale Constrained Optimization

Michaël Sdika

► **To cite this version:**

Michaël Sdika. A Fast Nonrigid Image Registration With Constraints on the Jacobian Using Large Scale Constrained Optimization. *IEEE Transactions on Medical Imaging*, 2008, 27 (2), pp.271-281. 10.1109/TMI.2007.905820 . hal-02073305

HAL Id: hal-02073305

<https://hal.science/hal-02073305>

Submitted on 1 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Fast Non Rigid Image Registration with Constraints on the Jacobian using Large Scale Constrained Optimization

Michaël Sdika

Abstract—This paper presents a new nonrigid monomodality image registration algorithm based on B-splines. The deformation is described by a cubic B-spline field and found by minimizing the energy between a reference image and a deformed version of a floating image. To penalize non invertible transformation, we propose two different constraints on the Jacobian of the transformation and its derivatives. The problem is modeled by an inequality constrained optimization problem which is efficiently solved by a combination of the multipliers method and the L-BFGS algorithm to handle the large number of variables and constraints of the registration of 3D images. Numerical experiments are presented on MR images using synthetic deformations and atlas based segmentation.

Index Terms—nonrigid registration, B-spline, Jacobian, multipliers method, multiresolution.

I. INTRODUCTION

THE problem of nonrigid registration is, given two images, to find a (nonlinear) function that maps each point in the reference image to a point in the floating image.

Nonrigid registration can be used to segment images using a prelabeled atlas [1][2][3], to estimate the motion in a sequence of images [4] and to compress and encode video [5]. It is also used for morphometry [6][7], or statistical analysis of medical images [8].

There are usually two ways to describe the transformation. The first solution is to use a dense non parametric model and estimate the position of each voxel after the transformation as in [9], [10], [11] or [3]. In this case, the transformation is often found by solving a nonlinear partial differential equation (PDE). The brain tissues are, for example, modeled as a material such as a linear elastic solid or viscous fluid that is subject to a deformation. The second solution is to model the transformation by a function of some parameters and estimate those parameters. In this case the transformation is known continuously (even between voxels), there are usually much fewer variables to estimate and the resolution of the transformation can be controlled independently of the image resolution. The smoothness of the parameterized function with respect to the voxels enables the analytical computation of the derivatives needed for deformation based morphometry or tensor based morphometry.

Manuscript received XXXX; revised XXXXX. This work was supported in part by the grant JF-2122-A from the National Multiple Sclerosis Society (PI : Dr Daniel Pelletier).

M. Sdika is with the Department of Neurology, University of California, San Francisco.

Our algorithm falls into the second category. The deformation is described by a linear combination of cubic B-splines and we estimate the coefficients of the combination to minimize the difference between the reference image and the deformed floating image. As the space spanned by B-splines is too general, we have to impose some constraints to find a realistic transformation between the images. For example, if the images are assumed to have the same structures, the transformation should be topology preserving.

As far as we know, the modeling of the deformation using B-splines was introduced by Szeliski and Coughlan in [12]. They proposed to regularize their deformation by adding a penalization term including first or second order derivatives of the deformation as done usually in PDE based registration.

In [13], Kybic *et al.* focused their work on the efficiency of the implementation and showed that B-splines were a good alternative to other parametrization such as wavelet or Fourier representation.

In [14], Sorzano *et al.* extended the usual regularization term based on the norm of the second derivatives taking into account the vectorial nature of the deformation. They obtained two terms based on the second derivative and, using the analytic properties of B-splines, they write them as two quadratic functions of the B-spline coefficients.

Rohlfing *et al.* [15] used a mutual information cost function. They penalized their cost function by adding the absolute value of the log of the Jacobian of the deformation or the square of the second derivative for each voxel. They used finite differences to compute the gradient of the Jacobian term.

In [16], Musse *et al.* used a block nonlinear Gauss-Seidel algorithm to minimize the energy between the images with the constraint that the Jacobian was continuously positive. Their method only applies to 2D images. They described their deformation by linear B-splines and, as they minimized the cost one node at a time, they showed that the constraints are reduced to two linear inequality constraints.

Noblet *et al.* extended the work of [16] in [17] to 3D images. They used interval arithmetic to find the maximum feasible step along their search direction to stay in the feasibility region where the Jacobian is continuously positive. This algorithm guarantees an invertible transformation between the images. However, it is restricted to linear B-spline as higher order B-splines would make the interval analysis part computationally prohibitive.

In [18], Rueckert *et al.* used the sufficient injectivity condition for B-spline of [19] to guarantee the invertibility. As

this condition is very restrictive, they describe their transformation with a composition of about forty invertible B-spline transformations. The topology preservation property is at the cost of the compactness and simplicity of the transformation representation.

Constraints on the Jacobian are also used in non parametric registration. In [20], Haber *et al.* bound the Jacobian during the registration. The optimization problem is solved using a log-barrier interior penalization method coupled with Gauss-Newton iterations. As they used a dense transformation, they have to pay special attention to the discretization of the constraints. Their dense representation also implies a larger number of unknowns and constraints.

In this paper, we propose to solve the registration problem by using a large scale constrained optimization algorithm. We also propose two constraints to penalize non invertible transformation. The Jacobian of the transformation is guaranteed to be above a positive threshold of all the voxels at the end of the algorithm and negative Jacobians are penalized between voxels.

In the first part, we will model the problem of non-rigid registration as a constrained optimization problem and describe its components, especially the constraints imposed on the transformation. We will then present the numerical algorithm used to solve our problem. We will finish with some numerical experiments on MR T_1 -weighted images of the brain to evaluate the different variants of the algorithm. A comparison is done with a classical regularization method and with the ITK [21] implementation of the Demons of Thirion [3].

II. THE NEW MODEL

We address in this article the problem of monomodality image registration. For the sake of concreteness, we consider 3D images but the generalization to any dimension is easy. We assume that the intensity of the voxels do not vary too much between the two images, and can be used directly to register the images. This assumption is usually satisfied after preprocessing steps such as histogram equalization or bias field removal for MRI images.

The transformation will be described by a linear combination of cubic B-splines uniformly placed on the image. The B-spline coefficients are found by solving an inequality constrained minimization. The cost function is data dependent and makes the transformation of the floating image fit the reference. The constraints (one per voxel) strongly penalize non invertible transformation.

Formally, the problem can be written as

$$\min_{\substack{g(c,x) \leq 0 \\ \forall x \in \mathcal{P}}} \frac{1}{|\mathcal{P}|} \sum_{x \in \mathcal{P}} \rho(I_f(t(c,x)) - I_r(x)), \quad (1)$$

where I_f is the floating image, I_r is the reference image, \mathcal{P} is the set of the voxels of the reference image, t is the transformation parametrized by the B-spline coefficients c to be estimated. $g(c,x)$ is the constraint on the transformation at the voxel x and will be given by the equations (2) or (3). The ρ function specifies the dissimilarity between the intensity

of the images. In the implementation of our algorithm, we use $\rho(x) = \frac{x^2}{2}$, but a robust M-estimation can be obtained if we use for example the Cauchy function $\rho(x) = \frac{\alpha^2}{2} \log(1 + (x/\alpha)^2)$ or the Geman-McClure function $\rho(x) = \frac{(x/\alpha)^2}{2(1+(x/\alpha)^2)}$. Multichannel registration can be done by considering I_r and I_f as vectors. A regularization term can also be added to the cost function if desired.

A. The Image Model

To evaluate the cost function, we only evaluate the reference image on the voxels, but we need the values of the floating image everywhere. So we define the floating image as a twice continuously differentiable function using interpolating cubic B-Spline:

$$I_f(x) = \sum_i a_i \beta(x - i),$$

where $i \in \mathbb{Z}^3$ and $\beta(x) = \beta(x_1)\beta(x_2)\beta(x_3)$ is the 3D tensor product of cubic B-spline. (Note that β will be indifferently used for the cubic B-spline or its tensor product). The interpolation coefficients a_i are efficiently found from the values at each voxel using the B-spline transform of [22]. Even though the evaluation of cubic B-spline is more expensive than a simpler model (nearest neighbor or linear interpolation), their smoothness makes them desirable from an optimization point of view. In practice, the computation time is reduced due to the reduced number of iterations (see also [23]).

B. The Parametrization of the Deformation

The deformation is also described by cubic B-splines, and thus is a twice differentiable function. The nodes are uniformly placed on the image. The transformation is described as:

$$t(c,x) = x + \sum_i c_i \beta(x/h - i).$$

The coefficients $c_i \in \mathbb{R}^3$ are the parameters to be estimated and $h \in \mathbb{N}^3$ is the spacing between the nodes.

C. The Search Space

As our transformation is modeled with cubic B-splines, it is twice continuously differentiable. For \mathcal{C}^1 functions, a necessary condition of invertibility is that its Jacobian is strictly positive everywhere in \mathbb{R}^3 . This condition, which is specified by an infinite number of inequalities, is not directly usable in this form in a numerical algorithm.

The first idea is to sample those constraints, that is to say to force the Jacobian to be above a given positive threshold for each voxel. The first constraint we propose for our model (equation (1)) is:

$$g(c,x) = g_1(c,x) = \varepsilon_d - J(c,x), \quad (2)$$

where x is a given voxel, c the B-spline coefficients and J the Jacobian of the transformation.

The g_1 constraint imposes the positivity of the Jacobian on all the voxels but lets the transformation free between the voxels. To overcome this problem, we propose to constrain the

Jacobian derivatives to be small when the Jacobian approaches its lower bound ε_d . The constraint proposed is based on the following lemma from [24]:

Lemma 1: Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a positive function, twice differentiable everywhere. Furthermore, let $f''(x) \leq M, M > 0 \forall x \in \mathbb{R}$, then, for any $x \in \mathbb{R}$,

$$\frac{1}{2}f'^2(x) \leq Mf(x).$$

This lemma establishes a necessary (and sufficient) condition between a positive function and its first derivative. Intuitively, one can understand that when a positive function approaches zero, so does its derivatives (see the proof in appendix I).

For a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ positive, if we apply the lemma 1 on the function obtained by fixing all but one variable, and summing the inequalities, we have :

$$\frac{1}{2} \|\nabla f(x)\|_2^2 \leq Mf(x).$$

We note that the J function is only piecewise twice differentiable, and that the lemma 1 cannot be applied. The second condition we propose for the registration is inspired by the lemma 1 to impose a relation between the Jacobian and its derivatives. This constraint is:

$$g(c, x) = g_2(c, x) = \frac{1}{2} \|\nabla J(x)\|_2^2 - \phi(J(x)), \quad (3)$$

where ϕ is a well chosen function.

A generalization of the condition imposed by the lemma 1 is permitted by the ϕ function, which enables more control on the variation of the Jacobian given its value. To keep the Jacobian in a given range, we just have to choose a function ϕ which is positive in this range and negative outside. For example, $\phi(z) = 4M(z - J_{min})(J_{max} - z)$ will only allow Jacobian values in $[J_{min}, J_{max}]$ at the voxels and bound its derivatives by $M(J_{max} - J_{min})^2$. When the Jacobian approaches J_{min} or J_{max} , its derivatives will be close to zero and the Jacobian will not go too far away from $[J_{max}, J_{min}]$ between the voxels. If a volume preserving registration is desired, one can take $\phi(z) = -(z - 1)^2$. This function will constrain all voxels to have Jacobian of 1 and null Jacobian derivatives.

To enforce the positivity of the Jacobian, we propose the following function:

$$\phi_{a,b,c,\varepsilon_d}(z) = \begin{cases} -a(z - \varepsilon_d)^2 & \text{if } z < \varepsilon_d \\ \frac{b(z - \varepsilon_d)^2}{(1+c(z - \varepsilon_d)^2)} & \text{otherwise.} \end{cases} \quad (4)$$

where the parameters a, b, c and ε_d are positive. $\phi_{a,b,c,\varepsilon_d}$ is an increasing \mathcal{C}^1 function. This function excludes the Jacobian values lower than ε_d on the voxels and when the Jacobian is close to ε_d , it forces its derivatives to be small, thus the Jacobian is less likely to be negative between voxels. Figure 1 shows a plot of $\phi_{a,b,c,\varepsilon_d}$ for $a = 100, \varepsilon_d = 0.01$ and several values of b and c .

Remark that for both g_1 and g_2 , the node spacing of the B-spline has an influence on the invertibility. Indeed, a larger h leads to a higher number of constraints per node.

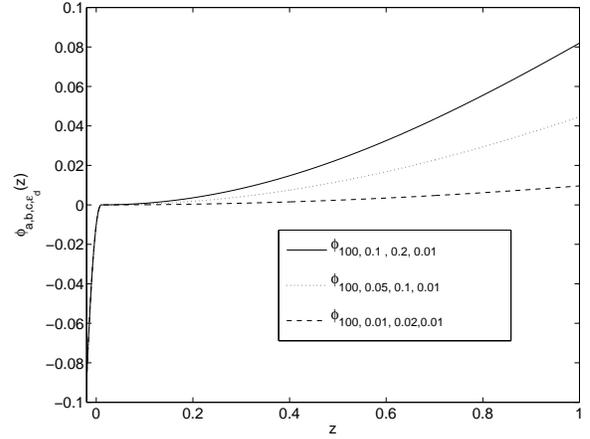


Fig. 1. The $\phi_{abc\varepsilon_d}$ function for $a = 100, \varepsilon_d = 0.01$ and several values of b and c .

III. THE OPTIMIZATION ALGORITHM

We now have to solve an inequality constrained optimization problem with a large number of variables and constraints. For example, to register 3D images of size $256 \times 256 \times 180$ with a node spacing of 6 voxels, the problem has 2×10^5 variables and 10^7 constraints. Note that for such a problem, the Hessian matrix cannot be stored in memory of current computers. Even storing only non zero elements in single precision would require 11 gigabytes.

We use the multipliers method which solves the constrained minimization problem by solving a sequence of unconstrained problem while estimating the Lagrange multipliers. The inner iterations will use the limited memory BFGS algorithm [25][26]. A multiresolution approach will provide a good starting point.

A. Multipliers Method

By writing the problem in the following way:

$$\min_{g(c) \leq 0} f(c),$$

we can define the augmented Lagrangian as

$$L_r(c, \mu) = f(c) + \sum_x \mu_x \tilde{g}(c, x) + \frac{r}{2} \sum_x (\tilde{g}(c, x))^2,$$

where $\tilde{g}(c, x) = \max(g(c, x), -\frac{\mu_x}{r})$, and μ_x are the Lagrange multipliers. Note that despite the max function, the augmented Lagrangian is continuously differentiable. The multipliers method consists of iteratively minimizing L_r with μ fixed then updating the μ variable and the penalty parameter r . This gives us the algorithm 1.

The starting point c_0 is provided by the multiresolution procedure. The inner minimization with fixed μ is done using a numerical algorithm (see III-B). In our implementation, r_0 was set to 10^4 .

The multipliers are updated using the first order formula which fits our requirement of low memory. This algorithm is

Algorithm 1 Multipliers Method

```

 $\mu = 0, c = c_0, c_{\text{prec}} = c_0, r = r_0$ 
while  $\|g(c)\|_\infty > C_{\text{stop}}$  do
   $c = \arg \min L_r(\cdot, \mu)$ 
   $\mu = \max(0, \mu + rg(c))$ 
  if  $\|g(c)\|_\infty > C_{\text{speed}}\|g(c_{\text{prec}})\|_\infty$  then
     $r = rC_{\text{incr}}$ 
  end if
end while

```

globally convergent, and the solution is obtained for a finite value of the penalty parameter. Note that for a simple exterior penalization, the solution is only obtained when r tends to infinity, making the problem ill-conditioned.

For more insight on the augmented Lagrangian and the multipliers method see [27][28][29][30][31].

The constants C_{stop} , C_{speed} and C_{incr} control, respectively, the stopping criterion, the speed of the decrease of the constraints violations and the increase of the penalty parameters. In our implementation, we took $C_{\text{speed}} = 0.4$, $C_{\text{incr}} = 1000$. The choice of C_{stop} , based on the two following propositions, will guarantee the positivity of Jacobian for all the voxels.

Proposition 1: At the end of the algorithm used with the g_1 constraint, the value of the Jacobian on the voxels is greater than $\varepsilon_d - C_{\text{stop}}$.

Proof: At the end of the algorithm, for any voxel x we have:

$$C_{\text{stop}} \geq g_1(c, x) = \varepsilon_d - J(c, x),$$

so

$$J(c, x) \geq \varepsilon_d - C_{\text{stop}}. \quad \blacksquare$$

Proposition 2: At the end of the algorithm used with the g_2 constraint and the $\phi_{a,b,c,\varepsilon_d}$ function, the value of the Jacobian on the voxels is greater than $\varepsilon_d - \sqrt{\frac{C_{\text{stop}}}{a}}$.

Proof: At the end of the algorithm, for any voxel x such that $J(c, x) < \varepsilon_d$, we have:

$$\begin{aligned} C_{\text{stop}} &\geq g_2(c, x) = \frac{1}{2} \|\nabla J(c, x)\|_2^2 + a(\varepsilon_d - J(c, x))^2 \\ &\geq a(\varepsilon_d - J(c, x))^2, \end{aligned}$$

from which we deduce:

$$J(c, x) \geq \varepsilon_d - \sqrt{\frac{C_{\text{stop}}}{a}}. \quad \blacksquare$$

In our implementation we took $C_{\text{stop}} = \frac{\varepsilon_d}{2}$ for the g_1 constraint and $C_{\text{stop}} = \frac{a\varepsilon_d^2}{4}$ for the g_2 constraint. Consequently, the Jacobian is always higher than $\frac{\varepsilon_d}{2}$ on the voxels. When we use g_2 , a large value for a gives a stronger constraint on negative Jacobian and a larger C_{stop} can be used to stop the algorithm earlier. However, a too large value for a will make the problem ill-conditioned. Taking $a = 100$ seems to be a good compromise.

B. Inner Iterations

The inner unconstrained minimization is solved using the limited memory BFGS algorithm with a non monotone line search.

The inverse BFGS formula is one of the most used quasi Newton formulas. At each iteration, a symmetric positive definite approximation of the inverse of the Hessian is updated and used to find the next iterate. The update is formed using a pair of vectors: the difference between the current point and the last iterate and the difference between the gradient of those two iterates. In the limited memory version, a predefined number of update pairs is stored and the matrix-vector product is formed from the list of update pairs. The Hessian matrix or its inverse is thus never explicitly formed (see [25][26]). The convergence of the algorithm and the validity of the BFGS updates are ensured by a non monotone version of the weak Wolfe line search of Lemaréchal (see [32][33][34]). Indeed, non monotone line searches improve the convergence of optimization algorithms by weakening the condition imposed to terminate the line searches.

C. Multiresolution

A simple coarse to fine approach will be used to provide a good starting point. A multiscale pyramid is built for the reference and the floating image. The problem is then solved for each level from the coarsest to the finest resolution. For each level, the starting point used is obtained by applying the EXPAND operator of [35] to the solution of the previous level.

This multiresolution approach is very important to reduce the computation time of the algorithm and to avoid local minima. Using a semi coarsening type of multiresolution (the resolution is increased in one direction at a time) gave us better results. Indeed the finer discretization in scale provided by semi coarsening reduce even more the chance to be trapped in a local minimum.

Note that accuracy is not necessary at intermediate scales. Except for the full resolution, the optimization is stopped after few iterations.

D. Numerical Issues

To be rigorous, the question of the constraint qualifications and the numerical stability of the discretization and the algorithm should be addressed. These questions are difficult to solve. However, the multipliers method avoids solving a KKT system and leads to a better conditioning. Then, the cubic B-splines used for the interpolation and derivatives of the floating image and the transformation as well as the spacing between the nodes help for the stability. Indeed, for each node, $4^3 h_1 h_2 h_3$ pixels have a direct influence during the registration.

IV. EFFICIENT IMPLEMENTATION OF THE AUGMENTED LAGRANGIAN AND ITS GRADIENT

We need to provide to the L-BFGS algorithm the value of the augmented Lagrangian and its gradient at each iteration of the inner minimization. In this paragraph, we give the

analytical expression for the gradient and how to implement it efficiently.

As pointed out in [13] for the unconstrained case, the gradient can be written as a sum of convolution with B-spline derivatives:

$$\frac{\partial L_r}{\partial c_i} = \sum_k \sum_x f_k(x) h^{-k} \beta^{(k)}(x/h - i),$$

where $k \in \mathbb{N}^3$ is the derivation order and $\beta^{(k)}$ a tensor cubic B-spline derivatives up to second order.

So the computation of the augmented Lagrangian and its gradient is efficiently done in three steps for each derivation order k :

- expansion of the deformation and its derivatives by upsampling the B-spline coefficients with a factor h followed by convolution with $h^{-k} \beta^{(k)}$,
- computation of the $f_k(x)$ for the different k for all the voxels,
- reduction by a convolution with $h^{-k} \beta^{(k)}$ followed by a downsampling with a factor h for the gradient computation.

The expression of the f_k functions for the two constraints proposed are given in the appendix II.

Note that we only need a full expansion of the B-spline field along the x_3 and x_2 directions. For the last dimension x_1 , we can expand one line at a time and save memory.

V. NUMERICAL EXPERIMENTS

The evaluation of nonrigid registration algorithms is a difficult task. Several metrics may be used to compare different registration methods (see for example [36]). In this paper, the registration is evaluated by measuring its ability to recover a synthetic transformation and to perform an atlas based segmentation of brain structures.

We compare the results of our algorithm with different regularization: without any constraints (NO), the g_1 constraint with a given ε_d ($g_1 \varepsilon_d$), the g_2 constraint with its parameters ($g_2 a, b, c, \varepsilon_d$), a standard bending energy penalization (BE λ), and the Demons algorithm of [3].

In the BE method, the cost function is completed by the term

$$E_\lambda(c) = \lambda \int_{\mathbb{R}^3} \sum_{i,j,k} \left(\frac{\partial^2 t_k}{\partial x_i \partial x_j}(c, x) \right)^2 dx,$$

which is a quadratic term in the B-spline coefficients. This penalization term and its derivatives are computed using the method described in [14].

For the Demons algorithm, we used a multiscale implementation provided by the ITK library [21] with a number of iterations of 100 as suggested in [36]. Experiments have been done with smoothing either the deformation field (Demons D σ) or the update field (Demons U σ).

A. An Inter Subject Registration Example

Our software without constraints, with the g_1 ($\varepsilon_d = 0.01$) and g_2 constraints (with $\phi_{100,2,1,0.01}$) as well as the Demons D $\sigma = 1$ algorithm have been used to register two different

subjects. T_1 weighted images acquired on a 3T scanner with a voxel size of $0.93mm \times 0.93mm \times 1mm$ were used. Before the registration, the brains have been extracted using BET[37]. Then the images have been smoothed with a Gaussian kernel ($\sigma = 0.7$) and histogram equalized to normalize the intensities and enhance the contrast.

Figures 2 and 3 shows the results of the registration for the slices 90 and 100. We show for each method the deformed image, the deformed image with superimposed contours of the reference and the deformation grid obtained by applying the transformation to a regular grid.

We can see that in general the Demons are visually not as good as the others. This is especially visible in the cortex at the bottom of the brain. The top of the ventricle, subject to a large deformation, does not match well either. Adding more iterations did not improve the results.

The output image with g_1 and g_2 constraints are very similar but the deformation grid of g_2 seems somewhat more regular than the one of g_1 .

Without constraints, the registration seems visually good and sometimes better than the results of g_1 or g_2 but this is at the cost of the topology preservation. For example, in the cortex, at the bottom left of the brain at slice 100, the output image matches better the contours of the reference. However, when we look at the deformation grid, we clearly see it contains folds.

For all the methods, the mean displacement of the voxels over the brain tissues is around 57.

B. Experiments with Synthetic Deformation

T_1 weighted images acquired on a 3T scanner with a voxel size of $0.93mm \times 0.93mm \times 1mm$ were used.

A set of 20 invertible deformations and 20 non invertible deformations has been synthetically generated and applied to an MR image. Then, on the original image, the bias field has been removed using the N3 algorithm of [38], a Gaussian noise of $\sigma = 30$ has been added (mean intensity of the white matter is 1194), a slight Gaussian smoothing has been applied ($\sigma = 0.7$) and a histogram equalization has been performed. Only the brain tissues have been used to compute the histogram. Next, each deformed image has been smoothed, histogram equalized and used as reference of the registration to the original image.

1) *The Deformations:* Our ground truth deformations are expressed as $d = d_n \circ \dots \circ d_2 \circ d_1$, compositions of deformation of the forms:

$$d_i(x) = x + \frac{r_i}{2} a_i s \left(\left\| \frac{x - c_i}{r_i} \right\|_2^2 \right),$$

where $a_i \in \mathbb{R}^3$, $c_i \in \mathbb{R}^3$, $r_i \in \mathbb{R}^{+*}$ and $s : \mathbb{R}^3 \rightarrow \mathbb{R}$. If we want to generate invertible transformations, we force each d_i to be invertible by choosing a_i small enough (see appendix III). An example of a deformed image is given in figure 4.

2) *Invertibility of the Output:* Our algorithm, with either of the two proposed constraints, guarantees that the Jacobian is strictly positive on all voxels. As the deformation we use in our registration algorithm is given by a parametric model, we

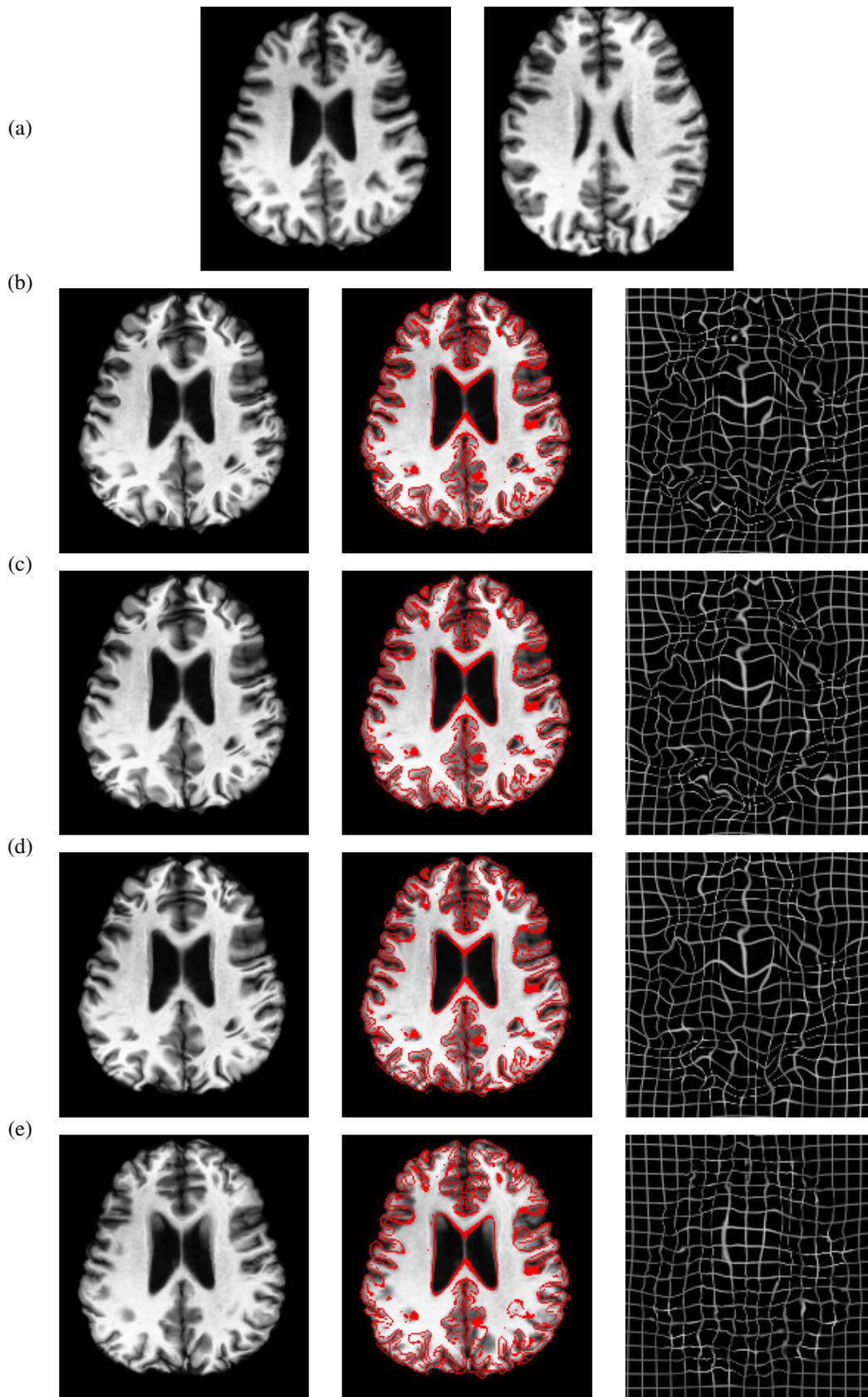


Fig. 2. Results for real subject registration. The first line (a) presents the reference (slice 90) and the floating images (slice 85). Lines (b), (c), (d) and (e) present the deformed image, the deformed image with superimposed contours of the reference as well as the deformation grid for the registration without constraints (b), with the g_1 constraint (c), with the g_2 constraint (d) and with the Demons algorithm (e).

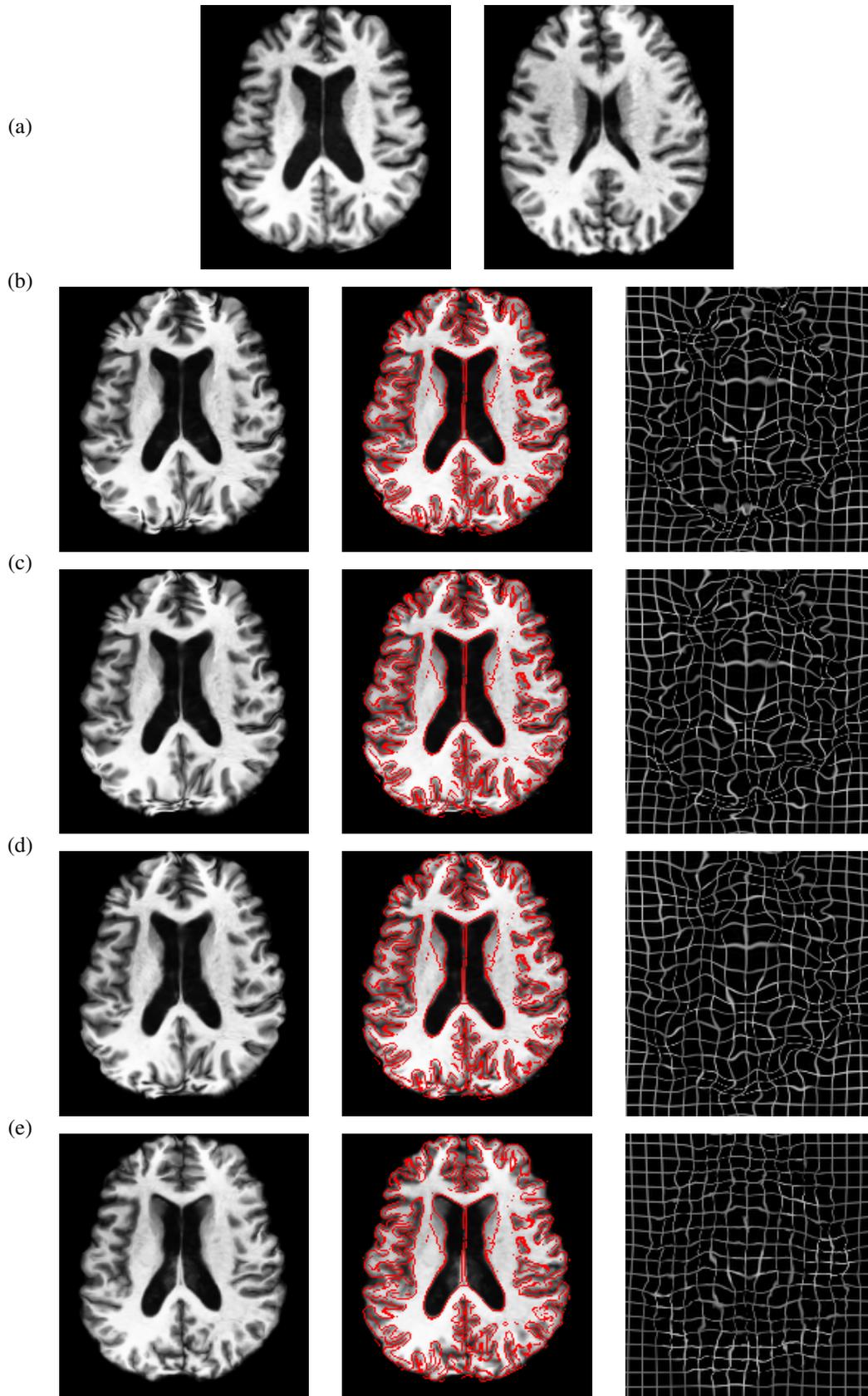


Fig. 3. Results for real subject registration. The first line (a) presents the reference (slice 100) and the floating images (slice 89). Lines (b), (c), (d) and (e) present the deformed image, the deformed image with superimposed contours of the reference as well as the deformation grid for the registration without constraints (b), with the g_1 constraint (c), with the g_2 constraint (d) and with the Demons algorithm (e).

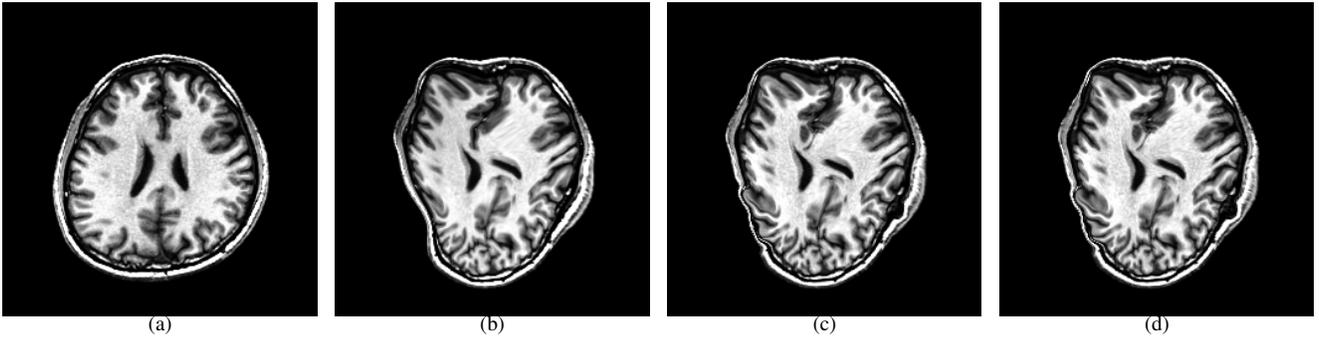


Fig. 4. Results for images with known simulated deformation: (a) floating image, (b) reference, (c) result with g_1 constraint, (d) result with g_2 constraint.

	Invertible			Non Invertible		
	$ J_{min} < 0 $	$E(J_{min})$	$E(err)$	$ J_{min} < 0 $	$E(J_{min})$	$E(err)$
NO	20	-3.171	1.08	20	-3.706	1.56
BE 0.001	20	-0.904	0.87	20	-0.844	1.33
BE 0.005	8	-0.003	0.73	18	-0.159	1.18
BE 0.05	0	0.224	1.43	1	0.203	2.13
g_1 0.01	18	-0.011	0.98	20	-0.023	1.44
g_1 0.10	0	0.068	0.99	0	0.059	1.46
g_1 0.01+BE 0.005	0	0.049	0.68	0	0.015	0.96
g_1 0.10+BE 0.005	0	0.079	0.66	0	0.080	0.97
g_2A	0	0.116	0.88	0	0.098	1.37
g_2B	0	0.008	0.95	0	0.006	1.48
g_2A +BE 0.005	0	0.087	0.51	0	0.068	0.88
g_2B +BE 0.005	0	0.008	0.61	0	0.005	0.94
Demons D 1.0			2.37			3.14
Demons U 3.0			2.11			2.41
Demons U 4.0			1.63			2.06
Demons U 5.0			1.85			2.33
Initial			59.68			59.79

TABLE I

RESULTS FOR SIMULATED TRANSFORMATION. WE HAVE REPORTED FOR EACH METHOD THE NUMBER OF REGISTRATIONS (OUT OF 20) PRODUCING NEGATIVE JACOBIANS ($|J_{min} < 0|$), THE MEAN OF THE MINIMUM VALUE OF THE JACOBIAN ($E(J_{min})$) AND THE MEAN OF THE ERROR BETWEEN THE OUTPUT AND THE GROUND TRUTH ($E(err)$). THE JACOBIANS ARE EVALUATED IN A $4 \times 4 \times 4$ FINER GRID AND THE ERROR IS EVALUATED ONLY ON BRAIN VOXELS. WE DISTINGUISH INVERTIBLE FROM NON INVERTIBLE GROUND TRUTH.

can analytically compute its value and derivatives everywhere and not only on the voxels.

To have an idea of the ability of the methods to preserve the positivity of the Jacobian even between voxels, we compute the analytical value of the Jacobian on a grid $4 \times 4 \times 4$ times finer than the original reference image. For each registration, we compute the minimum value of the Jacobian (J_{min}) of this grid. We didn't report the results for the Demons algorithm as one cannot compute the analytical value of the Jacobian in this case.

3) *Quality of the Registration*: For all the voxels, we can measure the error between the ground truth T_g and the output of the registration T :

$$e(x, T, T_g) = \|T(x) - T_g(x)\|_2.$$

As no information is available in the background, we only took into account the voxels inside the brain region \mathcal{B} given by BET [37]. To measure the quality of the registration, we measured the mean of the error

$$err(T, T_g) = \frac{1}{N_b} \sum_{x \in \mathcal{B}} e(x, T, T_g), \quad (5)$$

where N_b the number of voxels in the brain.

4) *Results*: The results of the simulations were reported in table I for the two sets of ground truth transformations (invertible or non invertible). The number of transformations with negative Jacobians, the mean of J_{min} and the mean of $err(T, T_g)$ are given for each method. The value of $E(err)$ before registration is 59.68 for the invertible simulation set and 59.79 for the non invertible one.

We ran our registration algorithm without constraints (NO), with the g_1 constraint with $\varepsilon_d = 0.01$ or $\varepsilon_d = 0.1$, with the g_2 constraint with $\phi_{100,0.01,0.02,0.01}$ (g_2A) or $\phi_{100,2,1,0.01}$ (g_2B) and with the bending energy penalization ($\lambda = 0.001, 0.005$ and 0.05). We also combined the g_1 or g_2 constraint with BE 0.005 (g_1+BE and g_2+BE).

We first noticed that with NO, we obtain negative Jacobians in all the cases. The registration without constraints does not give a good mean error compared to the other methods.

The standard regularization with a small penalization (BE 0.001 or 0.005) does improve the registration and reduces the number of negative Jacobians. However, an increase of the penalization coefficients to improve the invertibility leads to a strong reduction of the registration quality.

The Demons algorithms didn't give good results either.

We noticed however that Demons U gave better results than Demons D. This may be explained by the fact that, as pointed out in [39], Demons U approximates viscous fluid registration when Demons D approximates elastic registration and consequently, Demons U allows larger deformation.

The use of g_1 constraint slightly improves the registration results over NO. However, when $\varepsilon_d = 0.01$, the Jacobian becomes negative between the voxels almost everytime. A higher value or the addition of BE 0.005 is required to avoid negative Jacobians. Indeed, the addition of the BE produces more regular transformations, less likely to oscillate between voxels. Moreover, the addition of BE 0.005 to g_1 decreases significantly the mean registration error.

The g_2 constraint provides an improvement over NO as well. Similarly to g_1 , the addition of BE 0.005 improves the registration. The best results of the evaluation are obtained by $g_2A + BE 0.005$ with the lowest $E(err)$ and no negative Jacobians, even between voxels. You can notice that if, for g_2A , $E(J_{min})$ is much bigger than $\varepsilon_d = 0.01$, for g_2B , we have $E(J_{min}) \simeq \varepsilon_d$.

C. Experiments with Atlas Based Segmentation

1) *Presentation*: In this section, we evaluate the performance of the algorithm for segmentation of brain structures. For this application, preserving the topology allows us to avoid disconnect components, cavities or handles in the resulting segmentation.

We used the MR brain data set and their manual segmentation provided by the Center for Morphometric Analysis at Massachusetts General Hospital, available at <http://www.cma.mgh.harvard.edu/ibsr/>. There are 18 256x256x128 brain images with a resolution of 0.93x0.93x1.5 (8 images), 1.0x1.0x1.5 (6 images) or 0.83x0.83x1.5 (4 images).

The first image is used as the reference image. All the other images are registered to the reference and their segmentation mask mapped to the reference. Then for each structure of the segmentation, we measure the relative overlap between the ground truth S_g , manually segmented on the reference, and the segmentation given by the registration S_r :

$$RO(S_g, S_r) = 100 \times \frac{V(S_g \cap S_r)}{V(S_g \cup S_r)}.$$

A RO value of 100 means a perfect segmentation. We present in the table II, the mean of the relative overlap for five structures of the left part of the brain: the putamen (P), the caudate (C), the thalamus proper (TP), the lateral ventricle (LV), and the cerebral white matter (CWM). The global mean of RO over these five structures is given in the last column. The values of RO before registration are given at the last line of this table.

2) *Results*: We first see that using the Demons does not produce a good segmentation compared to the B-spline registration. In opposition to the previous experience, the bending energy penalization does not always improve the segmentation. If adding the BE penalization to NO, g_1 and g_2 improves the results for P and TP, this is not the case for C, LV and CWM.

A good segmentation can be produced with NO, BE, g_1 and g_2 but with the two constraints proposed, a good quality can be obtained while no Jacobians are encountered.

D. Computation Time

The computation time on a AMD processor 2.4 Ghz for images with 256x256x180 voxels and with a node spacing of 6x6x6 is around 4 minutes without constraints, 10 minutes with the g_1 constraint and 20 minutes with the g_2 constraint. The addition of the penalization to g_1 or g_2 reduces the CPU time by 30%. Demons D algorithm with $\sigma = 1.0$ and 100 iterations per level is about 10 minutes

VI. CONCLUSION

In this study, we have presented a nonrigid registration algorithm. The registration is modeled by a nonlinear optimization problem with nonlinear constraints to prevent the Jacobian to be negative. The optimization problem is solved using a combination of the multipliers method and the L-BFGS algorithm with a non monotone line search. The numerical resolution is efficient and does not require too much memory.

The transformation is modeled using cubic B-splines, making it intrinsically smooth, local, compact and fast to compute. The approach can however easily be applied to other parametrization.

Numerical experiments were performed with atlas based segmentation and recovery of a synthetic deformation and the proposed methods show good results.

Depending on the application, the invertibility of the transformation may not always be desirable "everywhere". For example, for inter subject registration, some topological changes may occur due to multiple sclerosis lesions or tumor. In these cases topological changes should be handled in a specific way but the topological constraints should be kept everywhere else. This issue will be addressed in a future work.

APPENDIX I PROOF OF THE LEMMA 1

Let (x, h) be two reals. By the Taylor-Lagrange theorem $\exists c_{x,h} \in]x, x+h[$:

$$f(x+h) = f(x) + f'(x)h + \frac{1}{2}f''(c_{x,h})h^2$$

By the positivity of f and the definition of M :

$$0 \leq f(x) + f'(x)h + \frac{1}{2}Mh^2$$

which is a positive second order polynomial in h , so its discriminant is negative:

$$f'^2(x) - 2Mf(x) \leq 0.$$

	$ J_{min} < 0 $	$E(J_{min})$	P	C	TP	LV	CWM	Mean
NO	17	-3.9014	64.6	64.8	71.3	77.5	71.9	70.0
BE0.001	17	-0.6453	68.3	65.8	73.7	75.8	69.2	70.5
BE0.005	5	0.0342	68.8	62.6	73.6	71.6	64.6	68.2
BE0.05	0	0.5291	66.4	56.7	71.5	61.6	56.4	62.5
$g_1 0.01$	17	-0.0305	65.0	65.4	71.7	78.0	72.0	70.4
$g_1 0.10$	0	0.0528	65.1	65.5	71.8	77.8	71.9	70.4
$g_1 0.01+BE 0.005$	0	0.0898	68.7	62.6	73.6	71.6	64.6	68.2
$g_1 0.10+BE 0.005$	0	0.1185	68.8	62.6	73.6	71.6	64.6	68.2
$g_2 A$	0	0.1489	67.4	64.0	72.8	73.8	68.9	69.4
$g_2 B$	0	0.0101	65.8	65.7	72.2	77.3	71.1	70.4
$g_2 A+BE 0.005$	0	0.1894	68.6	62.2	73.5	70.5	63.9	67.8
$g_2 B+BE 0.005$	0	0.0803	68.9	62.7	73.7	71.7	64.6	68.3
Demons D 1.0			61.0	61.3	69.4	73.0	65.7	66.1
Demons U 3.0			59.3	56.9	66.0	62.7	62.6	61.5
Demons U 4.0			62.2	57.8	69.0	62.0	59.8	62.2
Demons U 5.0			56.8	54.2	67.2	55.5	56.9	58.1
Initial			47.5	38.2	64.2	34.1	43.9	45.6

TABLE II

RESULTS FOR ATLAS BASED SEGMENTATION. WE HAVE REPORTED FOR EACH METHOD THE NUMBER OF REGISTRATIONS (OUT OF 17) PRODUCING NEGATIVE JACOBIANS ($|J_{min} < 0|$), THE MEAN OF THE MINIMUM VALUE OF THE JACOBIAN ($E(J_{min})$) AND THE MEAN RELATIVE OVERLAP FOR THE PUTAMEN (P), THE CAUDATE (C), THE THALAMUS PROPER (TP), THE LATERAL VENTRICLE (LV) AND THE CEREBRAL WHITE MATTER (CWM). THE LAST COLUMN IS THE MEAN RELATIVE OVERLAP OVER THESE FIVE STRUCTURES. THE JACOBIANS ARE EVALUATED IN A 4X4X4 FINER GRID.

APPENDIX II GRADIENT OF THE COST FUNCTION

In this appendix, we give the analytical expressions of the f_k functions of the section IV needed to compute the gradient of the augmented Lagrangian.

Let

$$v_{k_1 k_2 k_3} = \frac{\partial^k t(c, x)}{\partial x^k} = \frac{\partial^k x}{\partial x^k} + \sum_j c_j \prod_l h_l^{-k_l} \beta^{(k_l)} \left(\frac{x_l}{h_l} - j_l \right)$$

be the expansion of the B-spline coefficients using the kernel $\frac{\beta^{(k)}(x)}{h^k} = \frac{\beta^{(k_1)}(x_1)}{h_1^{k_1}} \frac{\beta^{(k_2)}(x_2)}{h_2^{k_2}} \frac{\beta^{(k_3)}(x_3)}{h_3^{k_3}}$ and

$$J = J(c, x) = \det(v_{100}, v_{010}, v_{001})$$

be the Jacobian of the deformation.

We also define the spatial derivatives of the Jacobian

$$\begin{aligned} J_1 = \frac{\partial J(c, x)}{\partial x_1} &= \det(v_{200}, v_{010}, v_{001}) \\ &+ \det(v_{100}, v_{110}, v_{001}) \\ &+ \det(v_{100}, v_{010}, v_{101}). \end{aligned}$$

$J_2 = \frac{\partial J(c, x)}{\partial x_2}$ and $J_3 = \frac{\partial J(c, x)}{\partial x_3}$ have similar expression.

The function associated with the zero order term is the data fitting term

$$f_{000} = \rho'(I_f(t(x)) - I_r(x)) \nabla I_f(t(x)).$$

For the terms of superior order we have two different expressions depending of the constraints we have chosen (g_1 or g_2). All those functions can be written

$$f_{k_1 k_2 k_3}(c, x) = \begin{cases} \tilde{f}_{k_1 k_2 k_3}(c, x) & \text{if } \mu_x + rg(c, x) > 0 \\ 0 & \text{otherwise} \end{cases}$$

If we note \times is the cross product of \mathbb{R}^3 , we have for the g_1 constraint (2):

$$\begin{aligned} \tilde{f}_{100} &= -(\mu_x + rg_1(c, x)) v_{010} \times v_{001} \\ \tilde{f}_{010} &= (\mu_x + rg_1(c, x)) v_{100} \times v_{001} \\ \tilde{f}_{001} &= -(\mu_x + rg_1(c, x)) v_{100} \times v_{010} \end{aligned}$$

For the g_2 constraint (3) if we define

$$\tilde{J}_i = (\mu_x + rg_2(c, x)) J_i$$

and

$$\tilde{\phi}' = (\mu_x + rg_2(c, x)) \phi'(J)$$

then we have:

$$\begin{aligned} \tilde{f}_{100} &= (v_{110} \times v_{001} - v_{101} \times v_{010}) \tilde{J}_1 \\ &+ (v_{020} \times v_{001} - v_{011} \times v_{010}) \tilde{J}_2 \\ &+ (v_{011} \times v_{001} - v_{002} \times v_{010}) \tilde{J}_3 \\ &- \tilde{\phi}' v_{010} \times v_{001} \\ \tilde{f}_{010} &= -(v_{200} \times v_{001} - v_{101} \times v_{001}) \tilde{J}_1 \\ &- (v_{110} \times v_{001} - v_{011} \times v_{001}) \tilde{J}_2 \\ &- (v_{101} \times v_{001} - v_{002} \times v_{001}) \tilde{J}_3 \\ &+ \tilde{\phi}' v_{001} \times v_{001} \\ \tilde{f}_{001} &= (v_{200} \times v_{010} - v_{110} \times v_{001}) \tilde{J}_1 \\ &+ (v_{110} \times v_{010} - v_{020} \times v_{001}) \tilde{J}_2 \\ &+ (v_{101} \times v_{010} - v_{011} \times v_{001}) \tilde{J}_3 \\ &- \tilde{\phi}' v_{001} \times v_{010} \\ \tilde{f}_{200} &= v_{010} \times v_{001} \tilde{J}_1 \\ \tilde{f}_{020} &= -v_{001} \times v_{001} \tilde{J}_2 \\ \tilde{f}_{002} &= v_{001} \times v_{010} \tilde{J}_3 \\ \tilde{f}_{011} &= v_{001} \times v_{010} \tilde{J}_2 - v_{001} \times v_{001} \tilde{J}_3 \\ \tilde{f}_{101} &= v_{001} \times v_{010} \tilde{J}_1 + v_{010} \times v_{001} \tilde{J}_3 \\ \tilde{f}_{110} &= -v_{001} \times v_{001} \tilde{J}_1 + v_{010} \times v_{001} \tilde{J}_2 \end{aligned}$$

APPENDIX III

GENERATING INVERTIBLE SYNTHETIC DEFORMATIONS

Let's define d_i by

$$d_i(x) = x + \frac{r_i}{2} a_i s \left(\left\| \frac{x - c_i}{r_i} \right\|^2 \right),$$

and α_s by

$$\alpha_s = \sup_{t \geq 0} \{t | s'(t^2)\}.$$

d_i will be invertible if we ensure that $\forall x \det(d'_i(x)) > 0$. Let $y_i = \frac{x-c_i}{r_i}$, we have:

$$\begin{aligned} \det(d'_i(x)) &= \det\left(I + s'\left(\|y_i\|_2^2\right) a_i y_i^T\right) \\ &= 1 + s'\left(\|y_i\|_2^2\right) a_i^T y_i \\ &\geq 1 - \left|s'\left(\|y_i\|_2^2\right)\right| \|y_i\|_2 \|a_i\|_2 \\ &\geq 1 - \alpha_s \|a_i\|_2 \end{aligned}$$

Thus, by choosing a_i such that $\|a_i\|_2 < \frac{1}{\alpha_s}$, the function d_i is invertible. For the function $s(t) = e^{-t}$ we used, we have : $\alpha_s = \frac{1}{\sqrt{2e}}$.

ACKNOWLEDGMENT

The author would like to thank Dr Daniel Pelletier, MD, Dr H el ene Ratiney, PhD and Alan Evangelista, BA for reviewing the manuscript and Dr Daniel Pelletier, MD for providing the funding for the realization of this work (grant JF-2122-A from the National Multiple Sclerosis Society).

REFERENCES

- [1] M. Miller, G. Christensen, Y. Amit, and U. Grenander, "Mathematical textbook of deformable neuroanatomies," 1993.
- [2] S. Warfield, A. Robatino, J. Dengler, F. Jolesz, and R. Kikinis, "Nonlinear registration and template driven segmentation," *Brain warping*, pp. 67–84, 1999.
- [3] J.-P. Thirion, "Image matching as a diffusion process: an analogy with Maxwell's demons," *Medical Image Analysis*, vol. 2, no. 3, pp. 243–260, 1998.
- [4] R. Szeliski and H.-Y. Shum, "Motion estimation with quadtree splines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1199–1210, 1996.
- [5] P. Moulin, R. Krishnamurthy, and J. Woods, "Multiscale modeling and estimation of motion fields for video coding," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1606–1620, 1997.
- [6] J. Ashburner, C. Hutton, R. Frackowiak, I. Johnsrude, C. Price, and K. Friston, "Identifying global anatomical differences: deformation based morphometry," *Human Brain Mapping*, vol. 6, pp. 348–57, 1998.
- [7] M. Schormann, T. Kraemer, "Voxel-guided morphometry ("vgm") and application to stroke," *IEEE Transactions on Medical Imaging*, vol. 22, no. 1, pp. 62–74, 2003.
- [8] J.-P. Thirion, S. Prima, G. Subsol, and N. Roberts, "Statistical analysis of normal and abnormal dissymmetry in volumetric medical images," *Medical Image Analysis (MedIA)*, vol. 4, no. 2, pp. 111–121, June 2000, version  electronique : <http://www.inria.fr/rrrt/RR-3178.html>.
- [9] R. Bajcsy and S. Kovacic, "Multiresolution elastic matching," *Comput. Vision Graph. Image Process.*, vol. 46, no. 1, pp. 1–21, 1989.
- [10] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, "A deformable neuroanatomy textbook based on viscous fluid mechanics," in *Proc. 27th Annual Conf. Inform. Sci. Syst., J. Prince and T. Runolfsson, Ed. Baltimore, MD: Department of Electrical Engineering, The Johns Hopkins University*, 1993, pp. 211–216.
- [11] G. E. Christensen, "Consistent linear-elastic transformations for image matching," in *IPMI '99: Proceedings of the 16th International Conference on Information Processing in Medical Imaging*. London, UK: Springer-Verlag, 1999, pp. 224–237.
- [12] R. Szeliski and J. Coughlan, "Spline-Based Image Registration, Tech. Rep. 94/1, April 1994.
- [13] J. Kybic and M. Unser, "Fast parametric elastic image registration," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1427–1442, November 2003.
- [14] C. S anchez Sorzano, P. Th evenaz, and M. Unser, "Elastic registration of biological images using vector-spline regularization," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 4, pp. 652–663, April 2005.
- [15] T. Rohlfing, C. Maurer, D. Bluemke, and M. Jacobs, "Volume-preserving non-rigid registration of MR breast images using free-form deformation with an incompressibility constraint," *IEEE Trans. Med. Imag.*, vol. 22, no. 6, pp. 730+, June 2003. [Online]. Available: citeseer.ist.psu.edu/rohlfing03volumepreserving.html
- [16] O. Musse, F. Heitz, and J.-P. Armspach, "Topology preserving deformable image matching using constrained hierarchical parametric models," *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 1081–1093, 2001. [Online]. Available: <http://lisiit.u-strasbg.fr/Publications/2001/MHA01>
- [17] V. Noblet, C. Heinrich, F. Heitz, and J.-P. Armspach, "3-d deformable image registration: a topology preservation scheme based on hierarchical deformation models and interval analysis optimization," *IEEE Transactions on Image Processing*, vol. 14, no. 5, pp. 553–566, may 2005.
- [18] D. Rueckert, P. Aljabar, R. A. Heckemann, J. Hajnal, and A. Hammers, "Diffeomorphic Registration using B-Splines," in *9th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2006)*, October 2006. [Online]. Available: <http://pubs.doc.ic.ac.uk/miccai06-diffeomorphic-reg/>
- [19] Y. Choi and S. Lee, "Injectivity conditions of 2d and 3d uniform cubic b-spline functions," *Graphical Models*, vol. 62, no. 6, pp. 411–427, 2000. [Online]. Available: citeseer.ist.psu.edu/choi00injectivity.html
- [20] E. Haber and J. Modersitzki, "Image registration with guaranteed displacement regularity," *Int. J. Comput. Vision*, vol. 71, no. 3, pp. 361–372, 2007.
- [21] L. Ibanez, W. Schroeder, L. Ng, and J. Cates, *The ITK Software Guide*, 1st ed., Kitware, Inc. ISBN 1-930934-10-6, <http://www.itk.org/ItkSoftwareGuide.pdf>, 2003.
- [22] M. Unser, A. Aldroubi, and M. Eden, "Fast B-spline transforms for continuous image representation and interpolation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 277–285, March 1991.
- [23] J. Kybic, "Elastic image registration using parametric deformation models," EPFL Thesis no. 2439 (2001), 180 p., Swiss Federal Institute of Technology Lausanne (EPFL), July 26, 2001.
- [24] A. Ambrosio, "Relation between positive function and its first derivative when its second derivative is bounded. [Online]. Available: <http://planetmath.org/encyclopedia/RelationBetweenPositiveFunctionAndItsFirstDerivativeWhenItsSecondDerivativeIsBounded.html>
- [25] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Programming*, vol. 45, no. 3, (Ser. B), pp. 503–528, 1989.
- [26] J. C. Gilbert and C. Lemar echal, "Some numerical experiments with variable-storage quasi-newton algorithms," *Math. Program.*, vol. 45, no. 3, pp. 407–435, 1989.
- [27] A. R. Conn, N. Gould, A. Sartenaer, and P. L. Toint, "Convergence properties of an augmented lagrangian algorithm for optimization with a combination of general equality and linear constraints," *SIAM J. on Optimization*, vol. 6, no. 3, pp. 674–703, 1996.
- [28] E. G. Birgin, R. A. Castillo, and J. M. Mart inez, "Numerical comparison of augmented lagrangian algorithms for nonconvex problems," *Comput. Optim. Appl.*, vol. 31, no. 1, pp. 31–55, 2005.
- [29] A. R. Conn, N. Gould, and P. L. Toint, "A globally convergent lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds," *Math. Comput.*, vol. 66, no. 217, pp. 261–288, 1997.
- [30] R. Andreani, E. G. Birgin, J. M. Mart inez, and M. L. Schuverdt, "On augmented lagrangian methods with general lower-level constraints," submitted for publication.
- [31] J. C. Gilbert, "Optimisation diff erentiable – th orie et algorithmes, syllabus de cours l'ENSTA," 2003.
- [32] C. Lemar echal, "A view of line-searches," *Optimization and optimal control*, no. 30, pp. 59–78, 1981.
- [33] L. Grippo, F. Lampariello, and S. Lucidi, "A nonmonotone line search technique for newton's method," *SIAM J. Numer. Anal.*, vol. 23, no. 4, pp. 707–716, 1986.
- [34] W. Sun, J. Han, and J. Sun, "Global convergence of nonmonotone descent methods for unconstrained optimization problems," *J. Comput. Appl. Math.*, vol. 146, no. 1, pp. 89–98, 2002.
- [35] M. Unser, A. Aldroubi, and M. Eden, "The L_2 -polynomial spline pyramid," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 4, pp. 364–379, April 1993.
- [36] V. Noblet, C. Heinrich, F. Heitz, and J.-P. Armspach, "Retrospective evaluation of a topology preserving non-rigid registration method," *Medical Image Analysis*, vol. 10, no. 3, pp. 366–384, 2006.
- [37] S. Smith, "Fast robust automated brain extraction," *Human Brain Mapping*, vol. 17, no. 3, pp. 143–155, November 2002.

- [38] A. Sled, J.G. Zijdenbos and A. Evans, "A nonparametric method for automatic correction of intensity nonuniformity in mri data," *IEEE Transactions on Medical Imaging*, vol. 17, no. 1, pp. 87–97, February 1998.
- [39] X. Pennec, P. Cachier, and N. Ayache, "Understanding the "demon's algorithm": 3D non-rigid registration by gradient descent," in *Proc. of 2nd Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI'99)*, ser. LNCS, C. Taylor and A. Colchester, Eds., vol. 1679. Cambridge, UK: Springer Verlag, September 1999, pp. 597–605.