Parallel Sampling with Stochastic Differential Equations for 3D Deformable Matching of Medical Images

Michel Salomon[1]†, Guy-René Perrin† and Fabrice Heitz‡
†LSIIT-ICPS, ‡LSIIT-MIV
Université Louis Pasteur, Strasbourg, UMR CNRS-ULP 7005
Pôle API, Boulevard Sébastien Brant 67400 Illkirch, France
salomon@lsiit.u-strasbg.fr

Submitted to PDPTA'2002

Abstract

Deformable inter-subject matching of 3D MR images remains a difficult problem due to the high dimension of both transformations and data. The matching problem is generally expressed as the minimization of a highly non-linear energy function, yielding a hard computationally expensive optimization problem.

In this paper, we consider a general fully parallel approach for the minimization of high dimensional, non-linear, energy function, based on the simulation of stochastic differential equations. The approach is naturally suited to massively parallel implementations, yields high quality registrations, and exhibits excellent relative speedup.

Keywords: 3D deformable image matching, global optimization, stochastic differential equations, massive parallelism.

1 Introduction

Image matching is an important preliminary task in many image processing problems like reconstruction of 3D information from 2D views or motion estimation from temporal sequences. The goal of image matching (also called image registration) is to geometrically align one image (the floating or source image) to another one (the reference or target image) so that voxels (or pixels in 2D) representing the same underlying structure may be superimposed.

In the present paper, we consider a hierarchical method for performing dense deformable inter-subject matching of 3D MR Images. The method uses a common framework for 3D image matching: it consists to find the transformation minimizing an energy (or cost) function that expresses the voxel similarity of the images to be aligned. The resulting global optimization problem is hard due to the high dimensionality of both transformations and data, but also because the energy function is highly non-linear and has many local minima. The registration of two 256^3 MRI typically involves the optimization of $200,000$ parameters, and takes several hours on a standard workstation. In this paper, we develop a comprehensive parallel implementation of the stochastic optimization algorithm, which enables a use of 3D deformable matching in clinical routine.

The paper is organized as follows. In section 2 we present the hierarchical deformable matching method [7]. In section 3 we describe the stochastic optimization algorithm based on the diffusion scheme proposed by Geman *et al.* [3], a simulated annealing like algorithm. The fourth section describes a comprehensive massively parallel implementation of the global optimization algorithm for our matching problem. Finally, section 5 is devoted to an experimental assessment of the computational performances of the parallel algorithm.

2 Dense hierarchical deformable matching

The matching method we consider [7] uses the following common framework, of estimating a displacement field d, by minimizing an energy function (or cost function) E corresponding to the sum of the squared differences metric:

$$E(d) = \frac{1}{|\Omega|} \int_{\Omega} |I(s) - J(s + d(s))|^2 ds \qquad (1)$$

where Ω is the bounded domain defined by the images ; $I(.)$ represents the reference image and $J(.)$ the floating image to be mapped onto image I; s designates a voxel coordinates in Ω. The displacement field is estimated according to:

$$d_{\min} = \arg\ \min\left[E\left(d\right)\right] \qquad (2)$$

[1] Supported by the Pluri-Formations Program "Analyse et Synthèse Multi-images", Ministry of National Education, Research and Technology, France (Grant 1997/2000).

where d belongs to an adequate Hilbert Space $\mathcal{H}$ of finite energy signals.

In the hierarchical method [7], E is not minimized directly on space $\mathcal{H}$; the optimization is performed over a sequence of increasing resolution nested subspaces $V_0 \subset V_1 \subset \cdots \subset V_l \subset \cdots \subset \mathcal{H}$, related to the theory of multiresolution signal decomposition [5]. Each $V_l, l \in \mathbb{Z}$ denotes the subspace containing the approximation of d at a given resolution 2^l.

Multiresolution approximation of 1D signals

Let us consider a sequence of multiresolution approximations in $L^2(\mathbb{R})$, constructed from a single compactly supported scaling function ϕ. Then the collection of functions $\left(\phi_i^l(x) = 2^{\frac{l}{2}}\phi(2^l x - i)\right)_{i\in\mathbb{Z}}$ forms a basis of a subspace V_l which is of finite dimension m_l ($i \in \{0,\ldots,m_l-1\}$). The approximation $\mathcal{A}^l f$ of a signal $f \in L^2(\mathbb{R})$ at resolution 2^l is obtained by its projection onto the basis of V_l:

$$\mathcal{A}^l f = A^l \bullet \left(\phi_0^l, \phi_1^l, \ldots, \phi_{m_l-1}^l\right)^T \tag{3}$$

where $A^l = \left(a_0^l, a_1^l, \ldots, a_{m_l-1}^l\right)$ is the parameter vector resulting from the projection.

Mallat [5] has shown that the parameter vector at resolution 2^{l+1} is related to the vector at resolution 2^l in the following way:

$$A^{l+1} = (\uparrow_2 (A^l)) * H \tag{4}$$

where $\uparrow_2$ is an operator that inserts zero between each element of A^l and H is a numerical convolution filter derived from the refinement equation.

3D deformation field modeling

The 3D deformation field $d : \Omega = [0,1]^3 \to \mathbb{R}^3$ is represented by a separable scaling function ϕ_{3D}, defined by: $\phi_{3D}(x,y,z) = \phi(x)\phi(y)\phi(z)$, where $\phi(x)$ is a 1D scaling function. Thus the basis of V_l becomes:

$$\Phi^l = \left(\phi_{i,j,k}^l(x,y,z) = 2^{\frac{3l}{2}}\phi(2^l x - i)\phi(2^l y - j)\phi(2^l z - k)\right) \tag{5}$$

where $i,j,k \in \{0,\ldots,m_l-1\}$

Finally, each displacement vector of d is projected at scale $l \in \mathbb{Z}$ onto V_l:

$$d^l(s) = d^l(x,y,z) = \begin{cases} d_x^l(x,y,z) = A_x^l \bullet {\Phi^l}^T \\ d_y^l(x,y,z) = A_y^l \bullet {\Phi^l}^T \\ d_z^l(x,y,z) = A_z^l \bullet {\Phi^l}^T \end{cases} \tag{6}$$

where $A_x^l = \left(a_{x_{i,j,k}}^l\right), A_y^l = \left(a_{y_{i,j,k}}^l\right)$ and $A_z^l = \left(a_{z_{i,j,k}}^l\right)$ are the parameter matrices (each of dimension m_l^3). The prolongation operator from scale l to $l+1$ is a straightforward extension of (4): the zero insertion operator and the filtering are easily extended to 3D vector fields.

Hierarchical optimization

As already explained, the matching method consists in estimating the 3D displacement field d that minimizes (1). This minimization is performed by considering a coarse-to-fine sequence of optimization problems, resulting from the successive restriction of d to subspaces $V_l, l \in \mathbb{Z}$. The energy function E (equation (1)) is minimized over each space V_l with respect to the parameter matrices. The energy function is highly non-linear and has many local minima, requiring global optimization schemes. Furthermore, we must take care of the number of parameter to be optimized. Considering the $\beta-$spline ϕ_1 scaling function ϕ [5], the scale range is 1 to 7 for 256^3 voxels MRI and the number of parameters to be optimized $var_\phi = \dim A_x^l + \dim A_y^l + \dim A_z^l = 3 \times {m_l}^3$. As can be seen below, over $700,000$ parameters have to be optimized at scale 6 (which is adopted in clinical applications).

l	m_l^3	var_ϕ	$var_{\phi,r}$
1	$1^3 = 1$	3	3
2	$3^3 = 27$	81	81
3	$7^3 = 343$	1029	930
4	$15^3 = 3375$	10125	5058
5	$31^3 = 29791$	89373	29346
6	$63^3 = 250047$	750141	198015
7	$127^3 = 2048383$	6145149	1436451

In practice, we take advantage of the large background area in the images in order to reduce the number of parameters ($var_{\phi,r}$). Thus, parameter corresponding to basis function whose supports $\Omega_{\phi_{i,j,k}^l}$ entirely cover the background are not estimated. In spite of this reduction, the computation burden is prohibitive on standard sequential workstation (several hours). To obtain computation times compatible with clinical routine, the parallelization of the global optimization algorithm must thus be considered.

3 Stochastic differential equations for global optimization

During the last decades many global optimization algorithms have been proposed: evolutionary algorithms [2] like genetic algorithms or evolution strategies, simulated annealing [4] and so on.

Simulated annealing is based on an analogy with the physical process of annealing. It consists in heating up a solid until it melts, then cooling it down until it crystallizes into a solid exhibiting a quasi perfect lattice and an energy (the function to be optimized) close to the global minimum. The simulation of the annealing process is related to stochastic sampling process from the Gibbs-Boltzmann distribution at temperature T. Well-known algorithms simulating the Gibbs-Boltzmann distribution are the Metropolis algorithm [6] and the Gibbs sampler. In this work we consider the following form of Langevin's stochastic differential equation [1, 3]:

$$dx_t = -\nabla E(x_t) \cdot dt + \sqrt{2 \cdot T_t} \cdot dw_t \qquad (7)$$

where E is the function to be minimized, it must be lipschitzian and verify $\lim_{\|x\| \to \infty} E(x) = +\infty$; T_t is the temperature at time $t \in \mathbb{R}^+$; w_t is a standard Brownian motion in $\mathbb{R}^N, N \in \mathbb{N}^+$; $X_t = x_t (= (x_{t,1}, x_{t,2}, ..., x_{t,N})) \in \mathbb{R}^N$ is an n-dimensional real vector. Equation (7) admits, at constant temperature T, the Gibbs-Boltzmann distribution as the limit density of X_t:

$$\lim_{t \to +\infty} P_T(X_t = x) = \frac{1}{Z_T} \exp\left(-\frac{E(x)}{T}\right) \qquad (8)$$

where $Z_T = \int_{\mathbb{R}^n} \exp\left(-\frac{E(x')}{T}\right) dx'$. If T decreases sufficiently slowly, the solution of (7) will converge to a distribution concentrated on the global minima of E. Conditions on E and T have been given by Geman *et al.* [3]. In particular it has been shown that the cooling schedule $(T_t)_{t \in \mathbb{R}^+}$ must satisfy:

$$\lim_{t \to +\infty} T_t \geq \frac{c}{\ln(2+t)} \qquad (9)$$

with c "sufficiently large".

Considering equation (7) as a sum of the following expressions:

$$\delta_1(t) = -\nabla E(x_t) \cdot dt \text{ and } \delta_2(t) = \sqrt{2 \cdot T_t} \cdot dw_t$$

gives a better understanding of the underlying idea. In fact, the first term $\delta_1(t)$ corresponds to a standard gradient descent, whereas the second term $\delta_2(t)$ introduces random perturbations controled by the temperature. Large values for T_t induce large perturbations, enabling the algorithm to escape from local minima. On the other hand, when the temperature decreases, the behavior becomes more and more a gradient descent.

The corresponding global optimization algorithm is obtained by numerical integration, discretizing equation (7) using the Euler-Cauchy method [1]:

$t = 0$ /* **Time initialization** */
$T_t \leftarrow T_0$ /* **Initial temperature** */
$\Delta P_t \leftarrow \Delta P_0$ /* **Initial time step** */
$X_t \leftarrow X_0$ /* **Random initial configuration** */
Repeat
  $\Delta w_{t,i} \leftarrow \mathcal{N}(0, \Delta P_t)$ /* **Brownian motion** */
  $x_{t+\Delta P_t, i} \leftarrow x_{t,i} - \frac{\partial E(x_t)}{\partial x_i} \cdot \Delta P_t + \sqrt{2 \cdot T_t} \cdot \Delta w_{t,i}$
  $T_{t+\Delta P_t} \leftarrow g(T_t)$
  $\Delta P_{t+\Delta P_t} \leftarrow h(\Delta P_t)$
  $t \leftarrow t + \Delta P_t$
Until *termination criterion met*

where $i \in \{1, ..., N\}, N \in \mathbb{N}^+$ and g, h are two strictly decreasing functions. Practically, the two functions are chosen in order to have a suitable exponential decrease of T_t and ΔP_t :

$$T_{t+\Delta P_t} = \alpha_T \cdot T_t ; \Delta P_{t+\Delta P_t} = \alpha_P \cdot \Delta P_t$$

with $t \in \mathbb{R}^+, 0 < \alpha_T < 1$ and $0 < \alpha_P < 1$.

4 Parallel sampling

The global optimization algorithm based upon Langevin's stochastic differential equation, described in the previous section, is fully parallel. Indeed, if the computation of the partial derivates of the energy function can be done in a fully independent way, then we can update simultaneously all the variables of the parameter vector to be optimized [8]. Thus, if we distribute the parameter vector in order to map each variable on one processor, we get a massively parallel implementation. Only communications to transmit the data needed to calculate the partial derivatives are requested. Let us notice that this parallelization has been used with success by Roysam *et al.* [8] for the reconstruction of patterns given noisy and/or incomplete data in medical imaging and radar imaging.

The relevance of the parallel sampling, based on the distribution of the partial derivates, is clearly related to the expression of the energy function gradient. In fact, the more the computations of the partial derivates are independent, the more efficient is the parallelization scheme. The gradient of the energy function is expressed in the next section.

Energy function derivatives

We derive the derivates for parameters $a^l_{x_{i,j,k}}$, where $i, j, k \in \{0, ..., m_l - 1\}$ (see equation (6)). The derivates with respect to $a^l_{y_{i,j,k}}$ and $a^l_{z_{i,j,k}}$ are calculated in the same way.

The derivates of E with respect to $a^l_{x_{i,j,k}}$ are:

$$\frac{\partial E\left(d^l(s)\right)}{\partial a^l_{x_{i,j,k}}} = \frac{1}{|\Omega|} \int_\Omega 2\left(J(s + d^l(s)) - I(s)\right) \frac{\partial\left(J\left(s + d^l(s)\right)\right)}{\partial a^l_{x_{i,j,k}}} ds$$

where $s = (x, y, z)^T$ and $d^l(s) = (d^l_x(s), d^l_y(s), d^l_z(s))^T$.

Introducing $s' = (x', y', z')^T = s + d^l(s)$ and using equation (6) yields the following expression:

$$\frac{\partial E\left(d^l(s)\right)}{\partial a^l_{x_{i,j,k}}} = \frac{1}{|\Omega|}\int_{\Omega} 2(J(s') - I(s))\frac{\partial J(s')}{\partial x'}\phi^l_{i,j,k}(s)ds$$

Furthermore as $\phi^l_{i,j,k}$ has $\Omega_{\phi^l_{i,j,k}} \subset \Omega$ for support, consequently:

$$\frac{\partial E\left(d^l(s)\right)}{\partial a^l_{x_{i,j,k}}} = \frac{1}{|\Omega|}\int_{\Omega_{\phi^l_{i,j,k}}} 2(J(s') - I(s))\frac{\partial J(s')}{\partial x'}\phi^l_{i,j,k}(s)ds$$

From this, we then derive the partial derivates matrices with respect to A^l_x, A^l_y and A^l_z:

$$\left(\frac{1}{|\Omega|}\int_{\Omega_{\phi^l_{i,j,k}}} 2(J(s') - I(s))\frac{\partial J(s')}{\partial x'}\phi^l_{i,j,k}(s)ds\right) \quad (10)$$

$$\left(\frac{1}{|\Omega|}\int_{\Omega_{\phi^l_{i,j,k}}} 2(J(s') - I(s))\frac{\partial J(s')}{\partial y'}\phi^l_{i,j,k}(s)ds\right) \quad (11)$$

$$\left(\frac{1}{|\Omega|}\int_{\Omega_{\phi^l_{i,j,k}}} 2(J(s') - I(s))\frac{\partial J(s')}{\partial z'}\phi^l_{i,j,k}(s)ds\right) \quad (12)$$

where $i, j, k \in \{0, ..., m_l - 1\}$.

A Massively parallel algorithm

The previous calculus shows clearly the relevance of the parallelization. As we can seen from (10), (11) and (12), the partial derivates computation on the sub-domains $\Omega_{\phi^l_{i,j,k}}$ may be performed independently and thus distributed on different processors. Hence, each partial derivates triplet $\left(\frac{\partial E(d^l(s))}{\partial a^l_{x_{i,j,k}}}, \frac{\partial E(d^l(s))}{\partial a^l_{y_{i,j,k}}}, \frac{\partial E(d^l(s))}{\partial a^l_{z_{i,j,k}}}\right)$ may be computed on the same processor. So, at scale l, we can use m_l^3 processors in a fully parallel scheme (in practice due to the parameter reduction $\frac{var_{\phi,r}}{3}$ processors). We give below the theoretical and practical number of processors that may be used simultaneously for a couple of 256^3 voxels 3D MRI.

l	Theory	Practice	l	Theory	Practice
1	1	1	5	29791	9782
2	27	27	6	250047	66005
3	343	310	7	2048383	478817
4	3375	1686			

Implementation

Obviously a SIMD parallel machine is the most suited to take profit of the large number of processors. However at the present time, MIMD machines dominate. All the experiments have been done using the SGI Origin2000 from Strasbourg University: it has 32 R10000 processors (195 MHz) and 20 R12000 processors (300 MHz), but we did not consider heterogeneous computing. Thus, we have studied the behavior of a massively parallel algorithm on this kind of parallel machine. The algorithm has been implemented in Fortran using OpenMP as parallel programming paradigm. It simply consists in a parallel loop dividing the work among processors.

5 Experimental results

The parallel algorithm has been applied on both 128^3 voxels and 256^3 voxels matching problems, using 3D MR Images (figure 1(a) and 1(d)) provided by the Institut de Physique Biologique (Strasbourg University Hospital).

Suitable tuning for $T_0, \alpha_T, \Delta P_0$ and α_P was found in some preliminary experiments, according to the parameters update number (*the termination criterion* - see section 3) that we set to 10 for each scale l.

Figure 1 shows a 3D visualization of the matching result for a particular 3D MRI couple. As we can see, the parallel sampling with stochastic differential equations is able to find parameters that align the images with good accuracy. As far as computation time is concerned (see below), for 128^3 voxels the parallel algorithm takes about 24 minutes on one R12000, whereas it takes less than 5 minutes on 16 R12000. The computational burden for 256^3 images is about 4 hours (for one R12000), compared to less than 40 minutes using 16 processors.

Images	IRM 128^3 voxels		IRM 256^3 voxels	
Scale	5		7	
l	Time (seconds)	Relative speedup	Time (seconds)	Relative speedup
R12000				
1	1468.26		13897.44	
8	341.58	4.30	3114.60	4.46
16	257.89	5.70	2293.80	6.06
R10000				
1	2095.35		20014.00	
8	487.75	4.30	4619.83	4.33
16	366.30	5.72	3446.88	5.81
24	329.25	6.36		
32	307.10	6.82		

To better assess the computational performances, let us compare the relative speedup (given in the previous tabular) to its upper bound given by Amdhal's law. For 128^3 voxels MRI, the stochastic sampling that can be parallelized on several processor ($l \geq 2$)

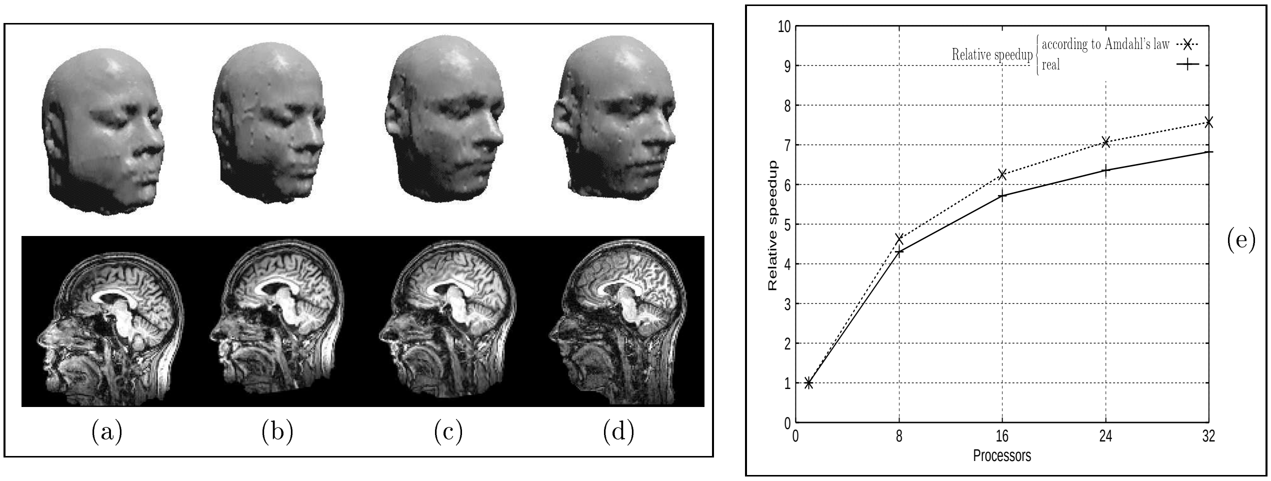

Figure 1: Non-rigid matching of 3D MR images from two different individuals (3D rendering): **(a)** is the source image and **(d)** is the target image. The results are shown after: deformable matching **(b)** at scale $l = 2$, and **(c)** at scale $l = 5$. Relative speedup curves (e).

requires 1877.15 seconds on one processor R10000, whereas the complete execution time is 2095.35 seconds. Thus :

- the fraction of the execution time that can be done in parallel is $p = \frac{1877.15}{2095.35} \approx 0.896$;
- the fraction of the execution time that is performed serially is $s = 1 - p \approx 0.104$.

Therefore, on eight processors for example, we have :

$$\text{Relative speedup} \leq \left(\frac{1}{s + p/8} = 4.63 \right) \tag{13}$$

which compares favorably to the experimental value (4.30). The good relatives speedups are highlighted by figure 1(e), considering 1, 8, 16, 24, and 32 R10000.

6 Conclusion

We have shown that a massively parallel implementation of stochastic differential equations-based global optimization is well suited for 3D deformable matching of medical images. It provides an efficient solution to the computational burden resulting from the high dimensionality of both transformations and data. CPU times of less than 40 minutes, compatible with clinical routine, have been obtained on a MIMD machine, to be compared with the several hours required on a standard workstation.

References

[1] F. Aluffi-Pentini, V. Parisi, and F. Zirilli. Global optimization and stochastic differential equations. *Journal of Optimization Theory and Applications*, 47(1):1–16, September 1985.

[2] T. Bäck. *Evolutionary algorithms in theory and practice*. Oxford University Press, 1996.

[3] S. Geman and C.R. Hwang. Diffusions for global optimization. *SIAM Journal on Control and Optimization*, 24:1031–1087, 1986.

[4] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, May 1983.

[5] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Analysis Machine Intelligence*, 2(7):674–693, 1989.

[6] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculation by fast computer machines. *Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

[7] O. Musse, F. Heitz, and J.-P. Armspach. Topology preserving deformable image matching using constrained hierarchical parametric models. *IEEE Trans. on Image Processing*, 10(7):1081–1093, July 2001.

[8] B. Roysam and M.I. Miller. Joint symbolic and stochastic inference via massively parallel architectures: application to medical imaging systems. Technical report, Rensselaer Polytechnic Institute, Troy, NY 12180, 1990.