

Parallel Non-Rigid Registration on a Cluster of Workstations

Radu Stefanescu, Xavier Pennec, Nicholas Ayache

INRIA Sophia, Epidaure, 2004 Rte des Lucioles, F-06902 Sophia-Antipolis Cedex

{Radu.Stefanescu, Xavier.Pennec, Nicholas.Ayache}@sophia.inria.fr

<http://www-sop.inria.fr/epidaure/Epidaure-eng.html>

Abstract: Over recent years, non-rigid registration has become a major issue in medical imaging. It consists in recovering a dense point-to-point correspondence field between two images and usually takes a long time. This is in contrast to the needs of a clinical environment, where usability and speed are major constraints, leading to the necessity of reducing the computation time from slightly less than an hour to just a few minutes. Another constraint is the usual unwillingness of healthcare organizations to invest in expensive high-performance computing solutions. Cluster computing proved to be a convenient solution to our computation needs, offering a large processing power at a low cost. Bi-processor workstation can be used simultaneously for parallel computation and individual day-to-day use and they are already present in many labs and hospitals. Our goal in this article is to provide a more usable tool by taking advantage of the available computation power.

Among the fast and efficient non-rigid registration algorithms, we chose the demons algorithm [1,2] for its simplicity and good performances. The parallel implementation decomposes the correspondence field into blocks, each block being assigned to one workstation of the cluster. We take advantage of the inherently regular structure of the algorithm that allows a nearly perfect static load balancing and also its locality, allowing to keep the amount of communication within a reasonable range. We obtained an acceleration of 11 by using 15 2GHz PC's connected through a 1GB/s Ethernet network and reduced the computation time from 40min to 3min30.

Keywords: Parallel computing, clusters, medical imaging, registration.

1. Introduction

Non-rigid registration is increasingly becoming an essential part of medical image processing algorithms involving more than one image. When given two different images of the same physical reality, it recovers a point-to-point correspondence between them.

Applications can be found in many domains of medical imaging. An often encountered medical procedure, the tracking of the evolution of a pathology uses as an important source of information medical images acquired at different moments in time. It is the task of the physician to compare the images and interpret changes. The operation is non-trivial when the images are in two dimensions and rather difficult when adding the difficulty of visualizing three-dimensional images. Furthermore, visual examination can produce qualitative, but not quantitative results, leaving the medical decision to a subjective evaluation. Non-rigid registration allows not only the identification of the correspondence between images, but it also gives quantitative and hence more objective results. Moreover, the registration is often only a part of a larger image processing chain, which includes interactive modules. This leads to the need of achieving the registration in a "human processing time", typically of one to two minutes.

Another domain of application we are interested in is Image-Guided Surgery (IGS). For example, in neurosurgery pre-operative guidance (using pre-operative images) allows the surgeon to select the best trajectory that safely invades the tissue. This step drastically reduces the operation time and the risks. Moreover, to complement the pre-operative planning, the surgeon uses an intra-operative guidance, in order to control his trajectory. Usually, image-guided surgery systems track surgical instruments during the operation and render the tracked devices within intra-operative images. However, this pre-operative planning, based on data acquired from the patient before the operation, tends to become obsolete during the operation due to fluid leaks and tissue removal. It is preferable to deform the pre-operative planning (thanks to registration) according to per-operative images, rather than entirely redoing it. For this application, computation times smaller than one minute are required.

Non-rigid registration is an ill-posed problem. Estimation of the matches between points often gives wrong results, leading to noisy correspondences. This is why a regularity hypothesis on the transformation is necessary. This is done by reducing the search space to regular transformations. In order to find the solution, an iterative optimization process is necessary and dealing with large deformations usually requires several tens of iterations.

A common approach of achieving shorter computation times is to sacrifice the algorithm performance by either using sub-sampled versions of the images or reducing the number of iterations. Our choice was to implement the algorithm on a parallel computer, which would allow us to keep the algorithmic performances intact, while drastically lowering computation time. This leads us to the second issue: Funding is generally unavailable in healthcare organization for purchasing expensive high-performance computing hardware. An obvious choice is therefore a cluster of networked personal computers. Besides offering large computation power at a low cost, a cluster of workstations has the advantage of versatility: it allows the use of its nodes as individual workstation in regular day to day use. Furthermore, a cluster of bi-processor PC's allows the simultaneous use of the cluster for parallel jobs and individual use, and such an environment us already present in many labs and clinical environments.

So far, several non-rigid registration algorithms have emerged. In 1981, Broit [5] used the linear correlation as a measure of similarity between the two images to match. Later, Bajcsy [6] differentiated this criterion and used a fixed fraction of its gradient as an external force to interact with a linear elasticity model. Thirion [7] proposed to consider non rigid registration as a diffusion process. He introduced in the images entities (demons) that push according to local characteristics. The forces he proposed were inspired from the optical flow equations.

Recently, Cachier [1] proposed an algorithm which replaces the optical flow equation from [7] with a gradient descent on an energy function combining a similarity term between the images and smoothness constraints. The transformation is represented by a dense correspondence field, that describes a specific match for each voxel. The optimization of the smoothness constraint is done through Gaussian filtering. Other algorithms use non-dense transformations: Ferrant [8] and then Rexilius [9] use a finite element bio-mechanical model in order to describe the deformation. Rueckert [10] uses the mutual information in order to find a deformation described using B-splines. These algorithms, although faster, suffer from the fact that they use non-dense transformations: they are generally unable to retrieve fine deformations.

We decided to use Cachier's demons algorithm [1] due to its combination of precision, robustness and relatively low computation time (30 to 45 minutes, depending on the images). In addition, the algorithm's rather regular structure makes it a good candidate for parallelization. We begin this paper by briefly describing the demons algorithm in section 2. In section 3, we establish the parallel decomposition and the parallel algorithm and we finish by addressing the more delicate problem of the Gaussian smoothing of images. Section 4 is dedicated to the presentation of experimental results.

2. The sequential algorithm

In the demons algorithm, the correspondence between a point p in the first image I and its match q in the other image J is characterized by the displacement (difference) $T(p)$ between the two points: $q = T(p) + p$. As the displacement is different in each point, $T(p)$ constitutes a *displacement field*. In the description below we will denote the composition of an image with a transformation as $(J \circ T)(p) = J(p + T(p))$. The algorithm attempts to estimate the regular displacement field T .

2.1. The optical flow

Let us suppose that we have a continuous set of images $I(t)$ evolving over time and we want to track a moving physical point $p(t)$ during that period of time. The optical flow is based on the assumption that the point p is observed with the same intensity in all images $I(t)(p(t)) = const$. We call this the intensity assumption. It characterizes the matching relationship defined above by the simple equality of the intensities of the physical point p in the two images. By differentiating the previous equation with respect to t , we get: $\nabla[I(t)](p)^T \cdot \frac{dp}{dt} + \frac{dI}{dt}(p) = 0$. We can therefore consider each physical point p to be in motion with speed v , where:

$$v = \frac{dp}{dt} = -\frac{\frac{dI}{dt}(p)}{\|\nabla[I(t)](p)\|^2} \nabla[I(t)](p)$$

Another assumption that can be made is that the norm of the displacement vectors between the two images to be registered is small when compared to the size of the kernel used to compute the derivatives. Therefore, the following approximation stands: $dI(p) = (J \circ T)(p) - I(p)$. Hence, the correction (CF) to add to the current displacement of a physical point p can be written as

$$CF(p) = \frac{I(p) - (J \circ T)(p)}{\|\nabla I(p)\|^2} \nabla I(p) \quad (1)$$

This yields a four-step iterative algorithm:

- Search for a residual correction CF of the current displacement field T .
- Smooth CF .
- Add $T=T+CF$.
- Smooth T and go to step 1 until convergence.

In the alternated optimization above, an identity correspondence field is used as initialization.

2.2. A multi-resolution approach

In order to deal with large deformations, the optimization loop is embedded into a multi-resolution approach. The algorithm begins by registering two small sub-sampled versions of the two images. When convergence is achieved, the resulted displacement field is over-sampled and serves as an initial estimate for registration at the next level.

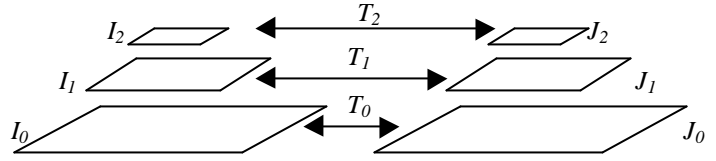


Figure 1: The registration is first done using the small sub-sampled versions of I and J (I_2 and J_2). The result T_2 is over-sampled and serves as an initial estimate for the registration between I_1 and J_1 .

2.3. Gaussian filtering

The Gaussian filtering is computed not by convolution, but by using its recursive implementation [4]. The main advantage of this algorithm is that, unlike the convolution, the computation time does not depend on the standard deviation of the Gaussian. This time depends linearly on the image size. In one dimension, the algorithm computes the coefficients of a fourth order recursive filter and applies them to the signal twice: forwards and backwards. In more dimensions, one can take advantage of the separability property of the Gaussian filter and do the filtering operation in each direction independently.

3. The parallel algorithm

The demons algorithm has a regular structure with three main "bricks": the search of the matches, the oversampling needed by the pyramidal approach and the regularization. It can be decomposed into sequential processing steps, and at each step each voxel is processed in a similar manner. This makes it a good candidate for a

parallelization using a *data decomposition*, rather than a *task decomposition*. We shall see now how each of these operations can be parallelized avoiding time-costly data redistributions.

3.1. Oversampled pyramid

Since the displacement field is distributed among the processors, the oversampling that is done each time the algorithm steps to another pyramid level must be done in parallel. Processors have to communicate to each other the values in the voxels that are at the border between two processor domains. Therefore, the parallel oversampling takes advantage from a block decomposition: the correction field would be broken in rectangular blocks, each of them being assigned to one processor.

3.2. Searching for the matches

The estimation of the match for each voxel is the simplest operation. In order to perform it for some voxel p , we only need to know the value of the correction field at the point p , the value of the input image I at voxel p and the input image J . Since images I and J are constant during the execution of the algorithm, each processor can memorize them entirely. The operation can be performed by each processor independently of the others, without any need for communication. Notice that each process must compute $J \circ T$ in its own domain, but since T can hold arbitrarily large displacements, each process must have access to the whole image J .

3.2. Parallel Gaussian filtering

The Gaussian is separable, so convolving a three dimensional image with a three dimensional isotropic Gaussian is equivalent to successively convolving the same image with three one-dimensional Gaussians. By adopting a block decomposition along one axis only, the filtering along two directions can be done within each block without communication. For the decomposition axis, one may benefit from the exponential decay of the Gaussian: It is common practice to consider the Gaussian to be null outside its $[m-3s, m+3s]$ interval (where we denoted with m the Gaussian's mean and with s its standard deviation). Therefore, when convolving a one-dimensional signal with a Gaussian, we can consider that the value of the filtered signal in some point x depends only of the values of the voxels of the initial signal within $x \pm 3s$. This leads to the

following simple algorithm (Figure 2): Each process sends its borders of width $3s$ (in gray in Figure 2) to its neighbors. Each process receives the neighbors' borders (in dashed grey) and adds them to its own domain, obtaining an extended domain. Each node filters its own extended domain and then throws away the received borders, in order to obtain a domain the size of the initial one. This algorithm has two disadvantages: Firstly, each process has to apply the filter to a domain that is larger than its own. Secondly, the amount of data sent through the network is proportional to the filter's standard deviation.

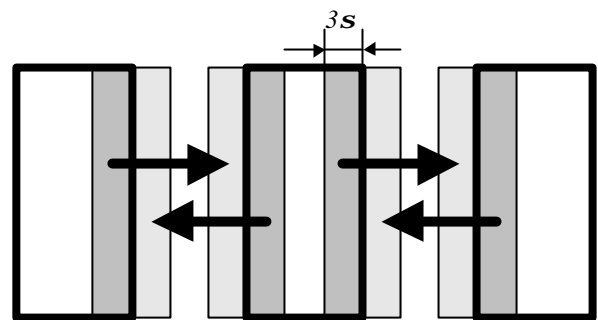


Figure 2: Each process sends its $3s$ wide borders to its neighbors and then filters the enlarged domain.

A second alternative is to directly parallelize the 4th order recursive implementation of the Gaussian proposed by Deriche [4], as follows. Let us consider the lines of the image along the block decomposition direction (we will call

them *scanlines*): they can be filtered independently of each other. Due to the recursive nature of the filter, computing the value of one voxel depends on the filtered version of the preceding voxel when filtering forwards, and of the following voxel when filtering backwards. This means that the filtering of *one single line* cannot be done in parallel.

However, different processors can deal with their parts of different lines simultaneously (Figure 3): At step 1, the left process begins processing its share of the first scanline. Meanwhile, processors 2 and 3 wait. Once processor 1 finished, it can pass on to processor 2 the contents of the 4 voxels (since the order of the filter is 4) that processor 2 needs in order to process its first voxel of its part of the first scanline. Process 1 filters its part of the second scanline while process 2 filters its part of the first scanline and process 3 does nothing. At the end of this step, process 1 passes the last 4 voxels of its part of the second scanline on to process 2, while the latter one sends the last 4 voxels of its part of the first scanline to the third process. This way, all the processes work simultaneously without filtering one scanline in parallel. The "process pipeline" however takes a number of steps equal to the number of processes before working at its full capacity. The full acceleration is achieved if the number of lines is much larger than the number of processes, which is usually true in a cluster of workstations.

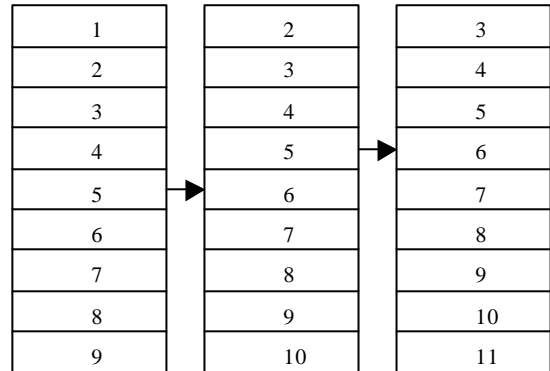


Figure 3: Parallel filtering of a 2D image of 10 lines with 3 processors. Inside each line, the step at which it is processed is given. At the end of step 5, processor 1 has just finished filtering its part of the fifth line, and is sending the last 4 voxels to processor 2. Meanwhile, processor 2 has finished filtering its part of the fourth line and is sending its last 4 voxels to processor 3 who has just finished filtering its part of the third scanline. At step 6, the three processors will filter their parts of lines 6, 5 and 4.

4. Results

We registered a couple of T1 MRI images of size 256x256x120. Two image pyramids of height 5 were built, and at each pyramid level the demons algorithm was iterated 40 times. The hardware platform consists in 15 Pentium IV PC's linked through 1GB/s Ethernet network. The computation times and the parallel acceleration are presented in figure 4. By using 15 processors, we obtained an acceleration of 11 (reducing the computation time from 40min to 3min30). This can be justified by the fact that some parts of the algorithm (creation of the image pyramids, computation of the gradients) are still sequential. For some processor configuration, the acceleration that was obtained was larger than the number of processors. We link this to the performance of the machines' memory. Since the algorithm uses large quantities of memory, cache misses are rather frequent when run sequentially. We believe that if the allocated memory falls below a certain limit, cache misses occur much less often, which largely improves the algorithm performance.

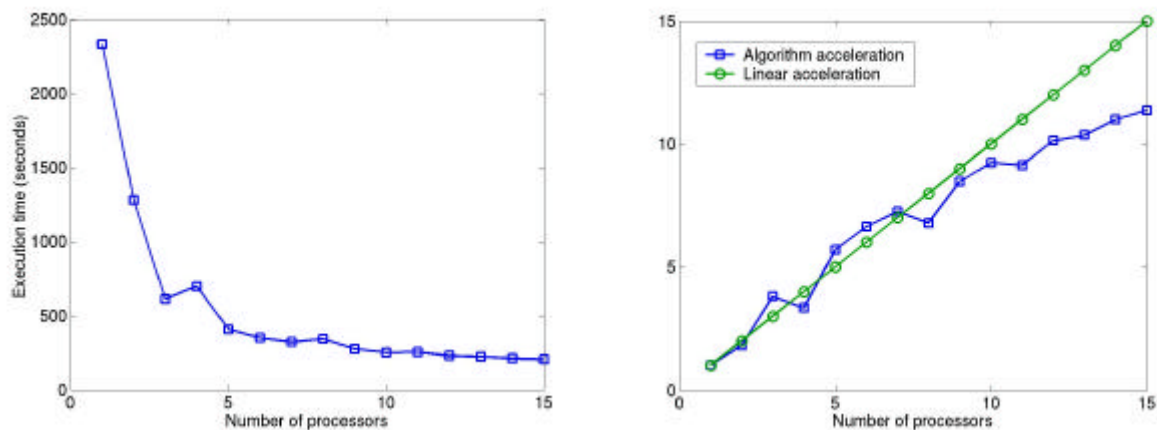


Figure 4

5. Conclusions

This paper is aimed at providing fast and more usable tools for non-rigid image registration by taking advantage of the available computation power already present in healthcare organizations. We present a distributed memory parallelization method for a non-rigid image registration algorithm. By using a block decomposition of the displacement field, the matching part of the algorithm can entirely be executed without communication. For the Gaussian smoothing, we present an efficient parallelization strategy which makes the computation time independent of the standard deviation of the Gaussian. Future work will consider means to accelerate the convergence of the algorithm, thus reduce the number of iterations and the computation time.

References

- [1] Pascal Cachier. Recalage non rigide d'images medicales volumiques – contribution aux approches iconiques et geometriques. PhD thesis, École Centrale des Arts et Manufactures, Paris, 2002
- [2] 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), LNCS 1679, Cambridge, UK, pages 597--605, September 1999. Springer Verlag.
- [3] P. Cachier and X. Pennec. 3D Non-Rigid Registration by Gradient Descent on a Gaussian-Windowed Similarity Measure using Convolutions. In Proc. of IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA'00), Hilton Head Island, South Carolina, USA, pages 182--189, June 2000. IEEE Computer society.
- [4] R. Deriche. Recursively Implementing the Gaussian and Its Derivatives. In Proc. Second International Conference On Image Processing, pages 263-267, Singapore, September 7-11 1992.
- [5] C. Broit. Optimal Registration of Deformed Images. PhD thesis, University of Pennsylvania, August 1981
- [6] R. Bajcsy and S. Kovacic. Multiresolution Elastic Matching. Computer Vision, Graphics and Image Processing, 46:1-21, 1989

- [7] J.-P. Thirion. Image Matching as a Diffusion Process: an Analogy with Maxwell's Demons. *Medical Image Analysis*, 2(3), 1998.
- [8] M. Ferrant, S.K. Warfield, A. Nabavi, F.A. Jolesz, and R. Kikinis. Registration of 3D Intraoperative MR Images of the Brain Using a Finite Element Biomechanical Model. In *MICCAI 2000*, pp. 19-28
- [9] J. Rexilius, S.K. Warfield, C.R.G. Guttmann, X. Wei, R. Benson, L. Wolfson, M. Shenton, H. Handels and R. Kikinis. A Novel Nonrigid Registration Algorithm and Applications. In *MICCAI 2001*, pp. 923-931.
- [10] Daniel Rueckert, Luke Sonoda, Derek Hill and David Hawkes. Non-Rigid Registration of contrast-enhanced MR mammography. *IEEE Transactions on Medical Imaging*, 18(8):712-721, 1999.