Diffeomorphic Demons Using ITK's Finite Difference Solver Hierarchy

Release 0.00

Tom Vercauteren[1,2], Xavier Pennec[1], Aymeric Perchant[2] and Nicholas Ayache[1]

June 6, 2007

[1]Asclepios Research Group, INRIA Sophia Antipolis, Sophia Antipolis, France
[2]Mauna Kea Technologies, Paris, France
Contact: `tom.vercauteren@m4x.org`

Abstract

This article provides an implementation of our non-parametric diffeomorphic image registration algorithm based on Thirion's demons algorithm. Within the Insight Toolkit (ITK), the demons algorithm is implemented as part of the finite difference solver framework. We show that this framework can be extended to handle diffeomorphic transformations. The source code is composed of a set of reusable ITK filters and classes. In addition to an overview of our implementation, we provide a small example program that allows the user to compare the different variants of the demons algorithm.

Contents

Forewords

This article is a companion paper to the authors MICCAI 2007 paper [15] entitled “Non-parametric diffeomorphic image registration with the demons algorithm”. It is intended to share the source code of our algorithm. As such it provides only basic information about the theory and does not present an evaluation of the method. The reader is thus invited to refer to [15] for the theoretical aspects and for an evaluation of the algorithm.

1 Introduction

Since Thirion’s seminal paper [13], the demons algorithm has become a popular method for the problem of intra-modality deformable image registration. The demons algorithm has successfully been used by several teams [16, 17] and an open source implementation of it is available in the Insight Toolkit [7]. The success of this method in the field of biomedical imaging can largely be explained by its efficiency. Thirion introduced *demons* that push according to local characteristics of the images in a similar way Maxwell did for solving the Gibbs paradox. The forces are inspired from the optical flow equations [2] and the method alternates between computation of the forces and their regularization by a simple Gaussian smoothing.

With the advent of computational anatomy and in the absence of a justified physical model of inter-subject variability, statistics on diffeomorphisms have become an important topic [1]. Diffeomorphic registration algorithms are at the core of this research field since they often provide the *input data*. They usually relies on the computationally heavy solution of a partial differential equation [3, 6, 8, 11, 12] or use very small optimization steps [5]. In [15], we proposed an efficient non-parametric diffeomorphic image registration algorithm based on an extension of the demons algorithm.

To the best of our knowledge, no diffeomorphic registration method has yet been integrated to the Insight Toolkit. The goal of this paper is to introduce the algorithm of [15] into ITK to provide an open source implementation of an efficient diffeomorphic image registration method.

2 Overview of the Algorithm

2.1 The Demons Algorithm

It has been shown in [4] that the demons algorithm could be seen as an optimization of a global energy. The main idea is to introduce a hidden variable in the registration process: correspondences. We then consider the regularization criterion as a prior on the smoothness of the transformation s. Instead of requiring that point correspondences between image pixels (a vector field c) be exact realizations of the transformation, one allows some error at each image point.

Given a *fixed image* $F(.)$ and a *moving image* $M(.)$, we end-up with the global energy:

$$E(c,s) = \frac{1}{\sigma_i^2}\,\mathrm{Sim}(F, M\circ c) + \frac{1}{\sigma_x^2}\,\mathrm{dist}(s,c)^2 + \frac{1}{\sigma_T^2}\,\mathrm{Reg}(s), \tag{1}$$

$$\mathrm{Sim}(F, M\circ s) = \frac{1}{2}\left\|F - M\circ s\right\|^2 = \frac{1}{2|\Omega_P|}\sum_{p\in\Omega_P}\left|F(p) - M(s(p))\right|^2, \tag{2}$$

where Ω_P is the region of overlap between F and $M\circ s$, σ_i accounts for the noise on the image intensity, σ_x accounts for a spatial uncertainty on the correspondences and σ_T controls the amount of regularization we need. We classically have $\mathrm{dist}(s,c) = \|c-s\|$ and $\mathrm{Reg}(s) = \|\nabla s\|$ but the regularization can also be modified to handle fluid-like constraints [4].

Within this framework, the demons registration can be explained as an alternate optimization over s and c. It can conveniently be summarized into the algorithm below:

Algorithm 1 (Demons Algorithm)**.**

- Choose a starting spatial transformation (a vector field) s
- Iterate until convergence:
  - Given s, compute a correspondence update field u by minimizing $E_s^{\mathrm{corr}}(u) = \|F - M \circ (s+u)\|^2 + \frac{\sigma_i^2}{\sigma_x^2}\|u\|^2$ with respect to u
  - If a fluid-like regularization is used, let $u \leftarrow K_{\mathrm{fluid}} \star u$. The convolution kernel will typically be Gaussian
  - Let $c \leftarrow s + u$
  - If a diffusion-like regularization is used, let $s \leftarrow K_{\mathrm{diff}} \star c$ (else let $s \leftarrow c$). The convolution kernel will also typically be Gaussian

In [14], we showed that a Newton method on $E_s^{\mathrm{corr}}(u)$ provided us with the folowing optimization step:

$$u(p) = -\frac{F(p) - M \circ s(p)}{\|J^p\|^2 + \frac{\sigma_i^2(p)}{\sigma_x^2}} J^{pT} \tag{3}$$

where we use the local estimation $\sigma_i(p) = |F(p) - M \circ c(p)|$ of the image noise and where $J^p = -\nabla_p^T(M \circ s)$ with a Gauss-Newton method, $J^p = -\frac{1}{2}\left(\nabla_p^T F + \nabla_p^T(M \circ s)\right)$ with the efficient second-order minimization (ESM) method of [10] and $J^p = -\nabla_p^T F$ with Thirion's rule. Note that σ_x then controls the maximum step length: $\|u(p)\| \leq \sigma_x/2$.

2.2 Newton Methods for Lie Groups

The most straightforward way to adapt the demons algorithm to make it diffeomorphic is to optimize (1) over a space of diffeomorphisms. This can be done as in [9, 10] by using an intrinsic update step

$$s \leftarrow s \circ \exp(u), \tag{4}$$

on the Lie group of diffeomorphisms. This approach obviously requires an algorithm to compute the exponential for the Lie group of interest. Thanks to the scaling and squaring approach of [1], this exponential can efficiently be computed for diffeomorphisms with just a few compositions:

Algorithm 2 (Fast Computation of Vector Field Exponentials)**.**

- Choose N such that $2^{-N}u$ is close enough to 0, e.g. $\max\left\|2^{-N}u(p)\right\| \leq 0.5$
- Perform an explicit first order integration: $v(p) \leftarrow 2^{-N}u(p)$ for all pixels
- Do N (not 2^N!) recursive squarings of v: $v \leftarrow v \circ v$

2.3 Diffeomorphic Demons

By plugging the above Newton method tools for Lie groups within the alternate optimization framework of the demons, we proposed in [15] the following non-parametric diffeomorphic image registration algorithm:

Algorithm 3 (Diffeomorphic Demons Iteration)**.**

- Compute the correspondence update field u using (3)
- If a fluid-like regularization is used, let $u \leftarrow K_{\text{fluid}} \star u$.
- Let $c \leftarrow s \circ \exp(u)$, *where* $\exp(u)$ is computed using Algorithm 2
- If a diffusion-like regularization is used, let $s \leftarrow K_{\text{diff}} \star c$ (else let $s \leftarrow c$).

3 Brief Note on the Implementation

Our implementation tries to follow the style and design of the Insight Toolkit. All our filter are N-dimensional and are templated over the important types such as the pixel types. In most cases we tried to divide the algorithm into meaningfull and reusable classes.

As shown in Algorithm 3, several blocks can be distinguished. We can first see that a method is required to compute the Lie group exponential of Algorithm 2. This algorithm takes a standard vector field as an input and provides a diffeomorphic vector field, which can be represented as a standard vector field, as an output. A natural choice was thus to implemented this exponential as an ITK image filter: the `ExponentialDeformationFieldImageFilter` class. This filter can easily be reused in a different setting such as to compute statistics on diffeomorphisms.

In order to ease the creation of the `ExponentialDeformationFieldImageFilter` class, several usefull and reusable filter such as the `DivideByConstantImageFilter` are also provided.

Within the Insight Toolkit, the demons algorithm is implemented as part of the finite difference solver (FDS) framework. Our implementation of the diffeomorphic demons is also built on top of this framework by implementing a specialized version of a `PDEDeformableRegistrationFilter`: the `DiffeomorphicDemonsRegistrationFilter` class. The most important modification we did to the FDS pipeline, is to include this exponentiation step within the `ApplyUpdate` function of our specialized `PDEDeformableRegistrationFilter` class.

In addition to these main classes, our submission also include a set a filters that are not fully part of the algorithm (e.g. `WarpJacobianDeterminantFilter`). These filters are meant to provide some statistics on the output of the algorithms. They should ease a quantitative comparison of the different variants of the demons algorithm.

Below is the list of classes, with brief descriptions, that we provide and use within our method:

- **itk::DiffeomorphicDemonsRegistrationFilter $<$ TFixedImage, TMovingImage, TDeformationField $>$:** Deformably register two images using a diffeomorphic demons algorithm
- **itk::DivideByConstantImageFilter $<$ TInputImage, TConstant, TOutputImage $>$:** Divide input pixels by a constant

- **itk::ESMDemonsRegistrationFunction $<$ TFixedImage, TMovingImage, TDeformationField $>$:** Fast implementation of the symmetric demons registration force
- **itk::ExponentialDeformationFieldImageFilter $<$ TInputImage, TOutputImage $>$:** Compute a diffeomorphic deformation field as the Lie group exponential of a vector field
- **itk::FastSymmetricForcesDemonsRegistrationFilter $<$ TFixedImage, TMovingImage, TDeformationField $>$:** Deformably register two images using a symmetric forces demons algorithm
- **itk::GridForwardWarpImageFilter $<$ TDeformationField, TOutputImage $>$:** Warp a grid using an input deformation field
- **itk::MultiplyByConstantImageFilter $<$ TInputImage, TConstant, TOutputImage $>$:** Multiply input pixels by a constant
- **itk::MultiResolutionPDEDeformableRegistration2 $<$ TFixedImage, TMovingImage, TDeformationField, TRealType $>$:** Framework for performing multi-resolution PDE deformable registration
- **itk::VectorCentralDifferenceImageFunction $<$ TInputImage, TCoordRep $>$:** Calculate the derivative by central differencing
- **itk::VectorLinearInterpolateNearestNeighborExtrapolateImageFunction $<$ TInputImage, TCoordRep $>$:** Linearly interpolate or NN extrapolate a vector image at specified positions
- **itk::WarpJacobianDeterminantFilter $<$ TInputImage, TOutputImage $>$:** Compute a scalar image from a vector image (e.g., deformation field) input, where each output scalar at each pixel is the Jacobian determinant of the warping at that location
- **itk::WarpSmoothnessCalculator $<$ TInputImage $>$:** Compute the harmonic energy of a deformation field

4 Users' Guide

From a user's point of view the most important file of our submission is the example application provided in `DemonsRegistration.cxx`. The goal of this example is to provide a command-line tool to perform an intra-modality deformable registration with a chosen variant of the demons. This tool works in both 2D and 3D and can trivially be extended to other dimensions.

The user can choose the input images, the variant of the demons that should be used and the type of output that should be stored. The image IO operations use standard ITK filters meaning that all file formats suported by ITK can be used.

Below is the list of options of the command-line tool:

- **-f/–fixed-image=STRING:** Fixed image filename - mandatory argument
- **-m/–moving-image=STRING:** Moving image filename - mandatory argument
- **-o/–output-image=STRING:** Output image filename - default: output.mha

- **-O/–output-field(=STRING):** Output field filename, optional argument- default: OUTPUTIMAGENAME-field.mha
- **-r/–true-field=STRING:** True field filename, this is for controlled experiments only where we want to compare the results of the algorithm with a known true field - default: not used
- **-n/–num-levels=UINT:** Number of multiresolution levels - default: 3
- **-i/–num-iterations=UINTx...xUINT:** Number of demons iterations per level - default: [10 10 10]
- **-s/–def-field-sigma=FLOAT:** Smoothing sigma for the deformation field at each iteration - default: 3
- **-g/–up-field-sigma=FLOAT:** Smoothing sigma for the update field at each iteration - default: 0
- **-l/–max-step-length=FLOAT:** Maximum length of an update vector (0: no restriction) - default: 2
- **-a/–use-vanilla-dem:** Use the vanilla demons instead of the diffeomorphic one
- **-t/–gradient-type=UINT:** Type of gradient used for computing the demons force (0 is symmetrized, 1 is fixed image, 2 is moving image) - default: 0
- **-e/–use-histogram-matching:** Use histogram matching (e.g. for different MRs)
- **-v/–verbose(=UINT):** Verbosity, if a verbose mode is used, the application will compute a set of statistics and write them to a text file - default: 0; without argument: 1
- **-h/–help:** Display an help message and exit

This command-line tool is used within a unit test triggered by CMake.

5 Conclusion

We have proposed an ITK implementation of our efficient non-parametric diffeomorphic registration algorithm. To the best of our knowledge, this is the first open-source implementation of a diffeomorphic registration tool within the Insight Toolkit. The design of our implementation tries to follow the design of ITK and thus provides templated N-dimensional filters. The code should be easily integrated to ITK and provide reusable blocks.

References

[1] Vincent Arsigny, Olivier Commowick, Xavier Pennec, and Nicholas Ayache. A Log-Euclidean framework for statistics on diffeomorphisms. In *Proceedings of the 9th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'06)*, pages 924–931, 2006.

[2] John L. Barron, David J. Fleet, and Steven S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994.

[3] M. Faisal Beg, Michael I. Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision*, 61(2), February 2005.

[4] Pascal Cachier, Eric Bardinet, Didier Dormont, Xavier Pennec, and Nicholas Ayache. Iconic feature based nonrigid registration: The PASHA algorithm. *CVIU — Special Issue on Nonrigid Registration*, 89(2-3):272–298, Feb.-march 2003.

[5] Christophe Chefd'hotel, Gerardo Hermosillo, and Olivier Faugeras. Flows of diffeomorphisms for multimodal image registration. In *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'02)*, pages 753–756, 2002.

[6] Gary E. Christensen, Richard D. Rabitt, and Michael I. Miller. Deformable templates using large deformation kinematics. *IEEE Transactions on Image Processing*, 5(10), October 1996.

[7] Luis Ibáñez, Will Schroeder, Lydia Ng, and Josh Cates. *The ITK Software Guide*. Kitware, Inc., 2 edition, 2005.

[8] Sarang C. Joshi and Michael I. Miller. Landmark matching via large deformation diffeomorphisms. *IEEE Transactions on Image Processing*, 9(8):1357–1370, August 2000.

[9] Robert Mahony and Jonathan H. Manton. The geometry of the Newton method on non-compact Lie-groups. *Journal of Global Optimization*, 23(3):309–327, August 2002.

[10] Ezio Malis. Improving vision-based control using efficient second-order minimization techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'04)*, April 2004.

[11] Stephen Marsland and Carole Twining. Constructing diffeomorphic representations for the group-wise analysis of non-rigid registrations of medical images. *IEEE Transactions on Medical Imaging*, 23(8):1006–1020, 2004.

[12] Michael I. Miller, Sarang C. Joshi, and Gary E. Christensen. Large deformation fluid diffeomorphisms for landmark and image matching. In Arthur Toga, editor, *Brain Warping*. 1998.

[13] Jean-Philippe Thirion. Image matching as a diffusion process: An analogy with Maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.

[14] Tom Vercauteren, Xavier Pennec, Ezio Malis, Aymeric Perchant, and Nicholas Ayache. Insight into efficient image registration techniques and the demons algorithm. In *Proceedings of Information Processing in Medical Imaging (IPMI'07)*, The Netherlands, July 2007. To appear.

[15] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache. Non-parametric diffeomorphic image registration with the demons algorithm. In *Proceedings of the 10th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'07)*, 2007. To appear.

[16] He Wang, Lei Dong, Jennifer O'Daniel, Radhe Mohan, Adam S. Garden, K. Kian Ang, Deborah A. Kuban, Mark Bonnen, Joe Y. Chang, and Rex Cheung. Validation of an accelerated 'demons' algorithm for deformable image registration in radiation therapy. *Physics in Medicine and Biology*, 50(12), 2005.

[17] Jocasta A Webb, Alexandre Guimond, Neil Roberts, Paul Eldridge, David W Chadwick, Jean Meunier, and Jean-Philippe Thirion. Automatic detection of hippocampal atrophy on magnetic resonance images. *Magnetic Resonance Imaging*, 17(8):1149–1161, April 1999.