Inverting deformation fields using a fixed point iteration scheme.

Release 0.10

Marcel Lüthi[1]

October 11, 2010

[1]Computer Science Department, University of Basel, Switzerland

Abstract

Being able to quickly compute the inverse of a deformation field is often useful in the context of medical image analysis. While ITK supports this functionality, the current algorithms are slow and do not always yield accurate results. In this paper we describe an ITK implementation of a fixed point algorithm for the approximate inversion of deformation fields, that was recently proposed by Chen et al. [1]. The algorithm has been shown to be both faster and more accurate than those currently implemented in ITK.

Contents

A commonly performed step in medical image analysis is to establish correspondence among a set of images, by means of a non-rigid registration algorithm. The resulting deformation fields are then used for a statistical analysis of the images (e.g. [4, 5]), or to transfer information from a annotated reference onto the target images [2, 3]. In this context it is often necessary to compute the inverse of the deformation field. In a recent paper, Chen et al. have proposed a fixed point iteration scheme to compute an approximate inverse of a deformation field [1]. They showed that their method is about 10 times faster than the currently implemented algorithms in ITK and, furthermore, yields more accurate results. The goal of this work is to make this algorithm available as an easy to use ITK filter.

1 Overview of the algorithm

Our implementation is based on the algorithm proposed by Chen et al. [1]. The idea is to construct a sequence of approximations to the true inverse deformation by employing a fixed-point iteration scheme. Let $\varphi : \Omega_A \to \Omega_B$ be a transformation from the image domain Ω_A to Ω_B. The deformation field u is the field of displacements of φ:

$$u : \Omega_A \to \mathbb{R}^d \tag{1}$$
$$x \mapsto \varphi(x) - x. \tag{2}$$

Denote by v the inverse deformation:

$$v : \Omega_B \to \mathbb{R}^d \tag{3}$$
$$x \mapsto \varphi^{-1}(x) - x. \tag{4}$$

v can be approximated using the following iterative fixed-point scheme:

$$v_0(x) = 0, \tag{5}$$
$$v_n(x) = -u(x + v_{n-1}(x)), n \in \mathbb{N}. \tag{6}$$

In [1] sufficient conditions for the convergence of Equation (5) are given. Our experience shows that for practical problems, where the deformation fields are rather smooth, the algorithm quickly converges and 20 iterations are usually enough to obtain a good approximation.

2 Experimental results

In this section we show results with our implementation. Following Chen et al. [1] we illustrate our method with a synthetic example. The deformation $u : \Omega_A \to \mathbb{R}^2$ at the point x is defined by the harmonic deformation:

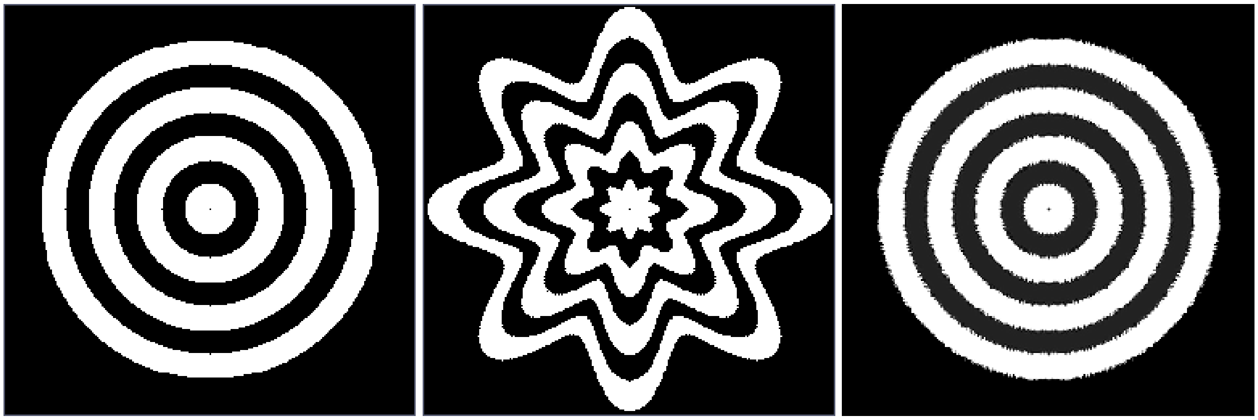
$$u(x) = \left(\frac{1}{1 + b \cos m\theta} \right) x$$

with $b = 0.2$, $m = 8$ and θ defining the polar angle of the point x. Figure 1a shows the effect of this deformation, applied to an image of concentric circles. We observe, that after warping the image once with the deformation u (Figure 1b) followed by the inverse u^{-1}, the original circle are well restored (Figure 1c). In the second example, we invert the deformation field obtained from registration of two images of the rat lung (Figure 2) using the Demons algorithm.[1] We warp the image once with the resulting deformation field and then back with its inverse. We observe that the image is accurately restored, even though the Demon's algorithm does not necessarily yield invertible deformation fields.

3 Using the ITK filter

The inversion algorithm is implemented as a subclass of `itk::ImageToImageFilter` and hence is straightforward to use. Its usage is the same as for the existing `itk::InverseDeformationFieldImageFilter`.

[1] We use the method given in the ITK example `DeformableRegistration2` for the registration.



(a) (b) (c)

Figure 1: A 2D toy example: The circles shown in (a) are warped using harmonic deformations. (b) shows the result of this deformation. The circles shown in (c) are the result of inverting the deformation and applying it to the warped image in (b).

Its main parameter is the number of iterations that are computed. Furthermore, the user needs to provide information such as origin, spacing and size of the output domain (i.e. Ω_A).

The following code example shows how the filter is used:

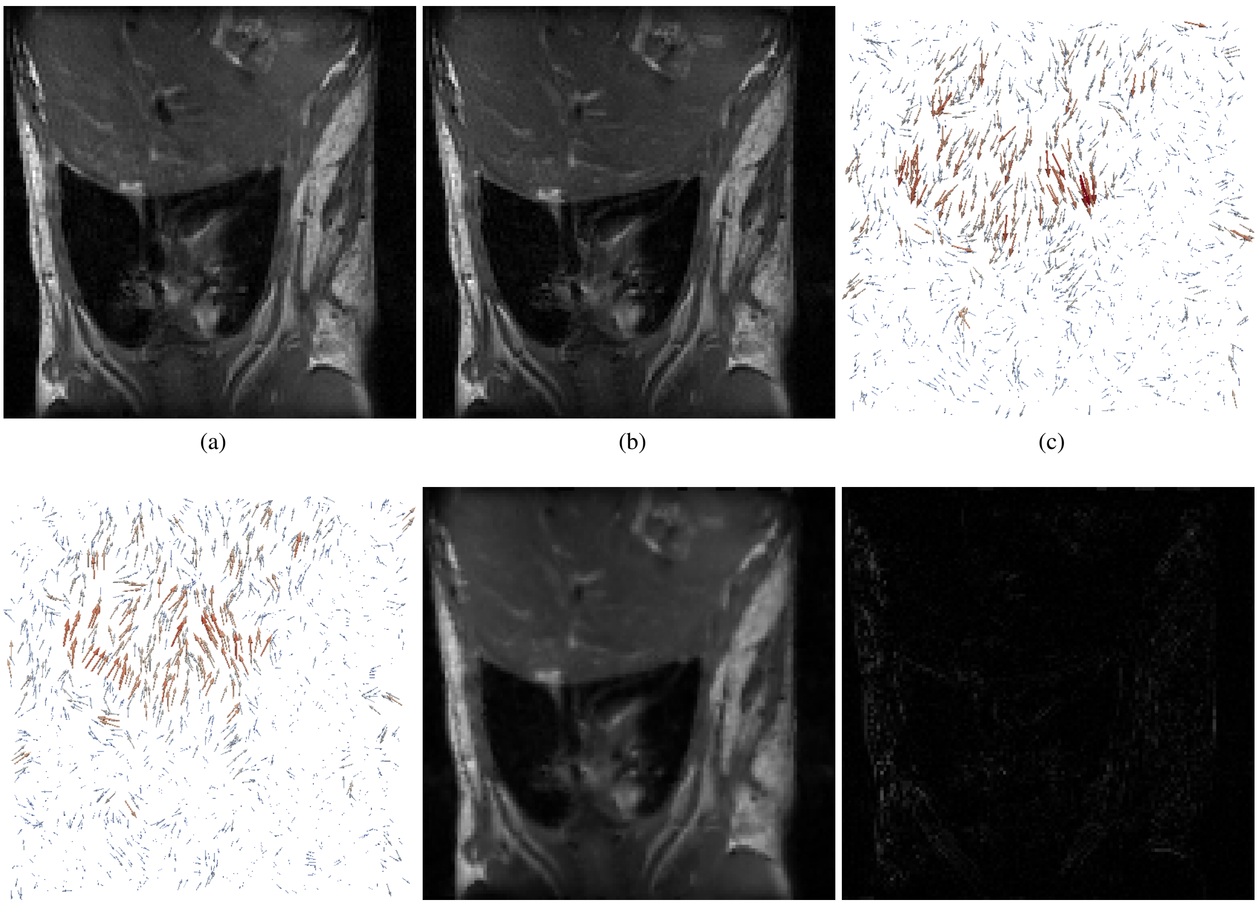
```
InputDFType::Pointer inputDf = reader->GetOutput();

typedef itk::FixedPointInverseDeformationFieldImageFilter<InputDFType,
                                                          OutputDFType> FPInverseType;
FPInverseType::Pointer inverter = FPInverseType::New();
inverter->SetInput(inputDf);
inverter->SetOutputOrigin(inputDf->GetOrigin());
inverter->SetSize(inputDf->GetLargestPossibleRegion().GetSize());
inverter->SetOutputSpacing(inputDf->GetSpacing());
inverter->SetNumberOfIterations(20);
inverter->Update();

OutputDFType::Pointer outputDf = inverter->GetOutput();
```

4 Conclusion

In this article we described an implementation of a fixed point iteration scheme for inverting deformation fields. It provides a faster and more accurate inversion than the currently available algorithms implemented in ITK. The algorithm can be used to obtain an approximate inverse, if the deformation field is not invertible.



(a) (b) (c) (d) (e) (f)

Figure 2: Fixed (a) and moving image (b) for the registration example. The deformation field that relates the two images is shown in (c) and its inverse in (d) (The deformations are scaled by the factor 3 for clarity of the illustration). (e) shows the result of warping the moving image with both the deformation field and its inverse. The difference shown in (f) that can be observed are mostly due to the effect of interpolation.

References

[1] M. Chen, W. Lu, Q. Chen, K. J Ruchala, and G. H Olivera. A simple fixed-point approach to invert a deformation field. *Medical Physics*, 35:81, 2008. (document), 1, 1, 2

[2] A. C Evans, W. Dai, D. L Collins, P. Neelin, and S. Marrett. Warping of a computerized 3-D atlas to match brain image volumes for quantitative neuroanatomical and functional analysis. In *Proceedings of SPIE*, volume 1445, page 236, 1991. (document)

[3] J.C. Gee, M. Reivich, and R. Bajcsy. Elastically deforming 3D atlas to match anatomical brain images. *Journal of computer assisted tomography*, 17(2):225, 1993. (document)

[4] U. Grenander and M. I Miller. Computational anatomy: An emerging discipline. *Quarterly of applied mathematics*, 56(4):694, 1998. (document)

[5] Daniel Rueckert, Alejandro F. Frangi, and Julia A. Schnabel. Automatic construction of 3D statistical deformation models using non-rigid registration. In *MICCAI '01: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, page 7784, London, UK, 2001. Springer-Verlag. (document)